Universidad de Sevilla

Facultad de Matemáticas

Grado en Matemáticas
Curso académico 2014-2015

Aplicaciones de la programación matemática a la selección de variables en algunos problemas de clasificación

Alfonso Peña Sánchez

Dirigido por: Justo Puerto Albandoz
Dpto. Estadística e Investigación Operativa
Fdo:

El alumno: Alfonso Peña Sánchez

Junio 2015, Sevilla

Índice

Resumen

Whether for how individuals are distributed in a population or recognize factors that cause diseases, classification is one of the fundamental objectives in science. Recognize patterns having a group of individuals (or objects) and makes them similar has various applications in other fields such as medicine, psychology, economics, marketing, engineering ...

Since centuries is studied the science of classification, known as taxonomy, in order to gain a better understanding of the world in which we live.

In this paper, Cluster analysis is proposed as a method for classifying individuals or entities characterized by a number of features.

Given a set of entities, Cluster Analysis aims at finding subsets, called clusters, which are homogeneous and/or well separated. As many types of clustering and criteria for homogeneity or separation are of interest, this is a vast field. A survey is given from a mathematical programming viewpoint.

However, this is not the main purpose of it. Cluster Analysis process goes through a pre-selection of variables that significantly affect the outcome of the analysis.

Clustering high-dimensional data is a difficult task if data contain variables with no relevant information. When those variables are not detected and discarded from the analysis, the analysis is blurred and biased by their presence. The problem becomes more and more important as the the number of variables increases.

Because of this, we need algorithms that select the important variables and eliminate those that adversely affect the clustering process.

This previous process is known as Variable Selection, and it is formulated as a linear optimization problem with binary variables representing the 0-1 decision of rejecting or selecting. The search of the subset of variables is a NP-hard problem (Kohavi, 1995), so that the use of metaheuristics obtains solutions without the need to explore the whole space of solutions. Therefore, it is useful to apply mathematical programming to address these problems. In this paper, we propose a mathematical programming viewpoint for Analysis Cluster and Variable Selection, besides certain models thereof.

Resumen

Tanto por conocer la distribución de individuos en una población o reconocer los factores que causan enfermedades, la clasificación es uno de los objetivos fundamentales de la ciencia. Reconocer los patrones que tienen un grupo de individuos (u objetos) y los hace similares tiene diversas aplicaciones en campos como la medicina, la psicología, la economía, marketing, ingeniería...

Durante siglos se ha estudiado la ciencia de la clasificación, conocida como Taxonomía, con el fin de obtener una mejor comprensión del mundo en que vivimos.

En este trabajo, el Análisis Cluster (Análisis de Conglomerados) se propone como método para la clasificación de personas o entidades que se caracterizan por una serie de variables.

Dado un conjunto de entidades, Análisis Cluster tiene como objetivo la búsqueda de subconjuntos, llamados clusters, que son homogéneas y/o se encuentren bien separados. Existen muchos tipos de agrupación y diversos criterios de homogeneidad o separación son de interés, haciendo de éste un campo amplio.

Sin embargo, este no es el propósito principal del mismo. El proceso de análisis de cluster pasa por un previo proceso de selección de las variables que afectan de manera significativa el resultado del análisis.

La agrupación de los datos de alta dimensión es una tarea difícil si los datos contienen variables con poca o ninguna información relevante. Cuando no se detectan y descartan el análisis se ve influenciado por su presencia. Este problema se vuelve más y más importante según se hace aumentar el número de variables. Es por esto que se necesitan algoritmos que seleccionen las variables importantes y eliminen aquellas que afecten negativamente el proceso de clasificación.

Este proceso previo se conoce como selección de variables, y se formula como un problema de optimización lineal con variables binarias que representan mediante 0-1 la decisión de rechazar o seleccionar.

La búsqueda del subconjunto de variables es un problema NP-duro (Kohavi, 1995), por lo que el uso de metaheurísticas obtiene soluciones sin la necesidad de explorar todo el espacio de soluciones. Por lo tanto, es útil aplicar la programación matemática para abordar estos problemas.

En este trabajo, se propone un enfoque basado en programación matemática para el Análisis Cluster y selección de variables, junto con ciertos modelos de ambos.

1. Introducción a los problemas de clasificación (Análisis Cluster)

¿Es posible identificar cuáles son las empresas en las que sería más deseable invertir?

¿Es posible identificar grupos de clientes a los que les pueda interesar un nuevo producto que una empresa va a lanzar al mercado?

¿Se pueden clasificar los animales de un rebaño según sus características productivas y aptitudes para la explotación ecológica (producción de leche, producción de carne, edad, enfermedades...) o las explotaciones ganaderas según su implicación en funciones no productivas?

¿Se pueden clasificar las bodegas de La Rioja en función de las características químicas y ópticas del vino que producen?

¿Es posible clasificar las estrellas del cosmos en función de su luminosidad?

Se trata fundamentalmente de resolver el siguiente problema: Dado un conjunto de individuos (de N elementos) caracterizados por la información de p variables v_j, ($j = 1, 2, ..., p$), se plantea el reto de clasificarlos de manera que los individuos pertenecientes a un grupo (al que llamaremos cluster, y siempre con respecto a la información disponible) sean tan similares entre sí como sea posible, siendo los distintos grupos entre ellos tan disimilares como sea posible.

Dentro de los métodos de Análisis Multivariante, el Análisis Cluster (también conocido como Análisis de Conglomerados, Taxonomía numérica o Reconocimiento de Patrones) es uno de los más recientes y tiene como objetivo la clasificación de individuos en grupos distintos, de forma que los perfiles de los objetos en un mismo grupo sean muy similares entre sí (homogeneidad, cohesión interna del grupo) y los de los objetos de grupos diferentes sean distintos (separación, aislamiento externo del grupo). Por comodidad, utilizaremos el término "$cluster$" para referirnos a estos grupos o conglomerados y "$clustering$" para referirnos a las respectivas agrupaciones.

Pertenece al igual que otras tipologías y que el Análisis Discriminante al conjunto de técnicas que tiene por objetivo la clasificación de los individuos. La diferencia fundamental entre el Análisis Cluster y el Discriminante reside en que en

el Análisis Cluster los grupos son desconocidos a priori y son precisamente los que queremos determinar; mientras que en el Análisis Discriminante, los grupos son conocidos y se pretende describir (si existen) las diferencias significativas entre ellos (sobre los que se observan p variables discriminantes), de forma que nos pueden ayudar a clasificar o asignar los individuos en los grupos dados.

Tiene una extraordinaria importancia en la investigación científica, en cualquier rama del saber, siendo la clasificación uno de los objetivos fundamentales de la ciencia. Sin embargo, junto con los beneficios del Análisis Cluster existen algunos inconvenientes. El Análisis Cluster es una técnica descriptiva y no inferencial. No tiene bases estadísticas sobre las que deducir inferencias estadísticas para una población a partir de una muestra, es un método basado en criterios geométricos y se utiliza fundamentalmente como una técnica exploratoria, descriptiva pero no explicativa. Las soluciones no son únicas, en la medida en que la pertenencia al cluster para cualquier número de soluciones depende de varios elementos del procedimiento elegido. Por otra parte, la solución depende totalmente de las variables utilizadas. La adición o destrucción de variables relevantes puede tener un impacto sustancial sobre la solución resultante.

Debido a esto último, es condición primordial en este tipo de estudio realizar una buena elección de las variables iniciales, así como también elegir una medida de homogeneidad o similaridad adecuada para la situación que se esté analizando. No existe una única medida de homogeneidad, ni tampoco es único el método de agrupar observaciones en distintos clusters. Es tarea del analista decidir qué medida y qué método son más adecuados según los datos de partida y los objetivos a conseguir con la agrupación.

Así pues, el objetivo es obtener clasificaciones (a las que también nos referiremos como clusterings), teniendo dicho análisis un marcado carácter exploratorio.

Para conseguir este objetivo, una vez establecidas las variables y los objetos a clasificar, el siguiente paso consiste en establecer una medida de proximidad o de distancia entre ellos que cuantifique el grado de similaridad entre cada par de objetos. Este paso, junto con la elección de distancia a definir entre dos clusters o entre un objeto y un cluster, serán los que más impacto tendrán sobre la solución final.

El proceso completo puede estructurarse de acuerdo con el siguiente esquema:

- Partimos de un conjunto de N individuos de los que se dispone de una información cifrada por un conjunto de p variables (una matriz de datos $N \times p$).
- Establecemos un criterio de similaridad para poder determinar semejanza de individuos entre sí.
- Escogemos un algoritmo de clasificación para determinar la estructura de agrupación de los individuos.
- Especificamos esa estructura mediante diagramas arbóreos, dendogramas u otros gráficos.

Los algoritmos para la agrupación se basan en estadística, matemática y ciencias de la computación. Dado un problema de análisis cluster, se pretende responder a los siguientes puntos:

1.Objetivo del agrupamiento (criterios).

2.La justificación de perseguir ese objetivo (axiomática).

3.Las restricciones a considerar (elección del tipo de agrupamiento).

4.Dificultad en llevar a cabo la agrupación (la cuestión de la complejidad).

5.Cómo debe hacerse la agrupación (diseño de algoritmos).

6.Significación del cluster obtenido (interpretación).

Una forma eficiente de abordar estas cuestiones es adoptar un punto de vista basado en programación matemática. Nos centraremos, por lo expuesto anteriormente, en los métodos que usan disimilitudes. El trabajo está organizado como sigue: El estudio del análisis cluster, bajo un punto de vista de programación matemática será visto en la siguiente sección. En la tercera sección se propone un marco para la selección de variables en los problemas de análisis cluster, donde la selección y la clasificación se realizan de forma simultánea.

2. Análisis Cluster y programación matemática

El problema de clasificación y agrupación se llevará a cabo mediante un punto de vista basado en la programación matemática. Esto permitirá la aplicación de técnicas y modelos conocidos a su resolución.

2.1. Componentes del Análisis Cluster

2.1.1. Pasos del Análisis Cluster

La mayoría de los métodos para el análisis de agrupamiento se basan en las disimilitudes (o similitudes) entre las entidades, valores numéricos que representan las discrepancias entre una entidad y otra, computados desde los datos previamente a relizar la agrupación. El esquema general para el clustering basado en disimilitudes es el siguiente:

a) Seleccionar una muestra $O = \{O_1, O_2, ..., O_N\}$ de N entidades entre las que se van a escontrar los clusters.

b) Matriz de datos: Medir p características de las entidades de O. Con esto se logra una matriz $N \times p$ de datos $B = (O_{ij})$.

$$B = \begin{pmatrix} O_{11} & O_{12} & ... & O_{1p} \\ O_{21} & O_{22} & ... & O_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ O_{N1} & O_{N2} & ... & O_{Np} \end{pmatrix}$$

c) Disimilitudes. Computar de la matriz B una matriz $N \times N$ llamada $D = (d_{kl})$ de disimilitudes entre las entidades. Dichas disimilitudes a menudo satisfacen las propiedades: $d_{kl} \geq 0, d_{kk} = 0, d_{kl} = d_{lk}$ para $k, l = 1, 2, ..., N$. No tienen por qué satisfacer la desigualdad triangular (es decir, no tienen por qué ser distancias).

$$D = \begin{pmatrix} d_{11} & d_{12} & ... & d_{1N} \\ d_{21} & d_{22} & ... & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & ... & d_{NN} \end{pmatrix}$$

d) Restricciones. Elegir el tipo de clustering o agrupación deseada. También especificar restricciones sobre los grupos obtenidos durante el agrupamiento, como máxima cardinalidad, peso de las entidades para un cluster concreto...

e) Criterios. Escoger un criterio (o más) para expresar la homogeneidad y/o la separación de los grupos en el clustering.

f) Algoritmo. Escoger o diseñar un algoritmo para los problemas definidos en d) y en e). Obtener el software correspondiente.

g) Computación. Aplicar el algoritmo escogido a la matriz D, obteniendo los grupos (clusters) del clustering elegido por las restricciones de d).

h) Interpretación. Aplicar pruebas formales o informales para seleccionar el mejor clustering de los obtenidos en el paso anterior. Describir los grupos obtenidos mediante las entidades que poseen y estadística descriptiva. A continucación, proceder a una interpretación de los resultados.

Los pasos de a) a c) corresponden a un punto de vista estadístico para el problema de clustering. Este trabajo se centrará en los pasos entre d) y g), que corresponden al punto de vista de la programación matemática.

Hay que hacer varias observaciones sobre esto:

- En primer lugar, las disimilitudes pueden ser computadas mediante otro procedimiento diferente a una matriz de datos B, como por ejemplo, al comparar secuencias biológicas.
- En segundo lugar, Análisis Cluster no es la única forma de estudiar las disimilitudes o distancias entre entidades en el campo del Análisis de Datos. Métodos como el Análisis de Componentes Principales o el Análisis Factorial también pueden ser utilizados. Sin embargo, este trabajo se centrará en el Análisis Cluster.
- Tercero, en lugar de computar disimilitudes, se puede proceder a un clustering directo en la matriz B, pero los clusters obtenidos de esta forma deben ser interpretados en términos conceptuales.

2.1.2. Tipos de $clustering$ (agrupaciones)

Los algoritmos para el análisis de conglomerados están diseñados para encontrar varios tipos de clustering o agrupaciones, como son:

1)Subconjunto C de O

2) Partición $P_M = \{C_1, C_2, ..., C_M\}$ de O en M clusters, verificando:
2.1) $C_j \neq \emptyset$ para $j = 1, 2, ...M$
2.2) $C_i \cap C_j = \emptyset$ para $i, j = 1, ..., M$ y $i \neq j$
2.3) $\bigcup_{j=1}^{M} Cj = O$

3)Empaquetamiento $Pa_M = \{C_1, C_2, ..., C_M\}$ de O con M clusters, verificando lo mismo que 2) excepto la condición 2.3).

4) Cubrimiento $Co_M = \{C_1, C_2, ..., C_M\}$ de O por M clusters, verificando lo mismo que 2) excepto la condición 2.2).

5) Jerarquía $H = \{P_1, P_2, ..., P_q\}$ de $q \leq N$ particiones de O. Es un conjunto de particiones $P_1, P_2, ..., P_q$ tales que si $C_i \in P_k, C_j \in P_l$ y $k \geq l$ entonces $C_i \subset C_j$ ó $C_i \cap C_j = \emptyset$ para todo i, j con $i \neq j, \quad k, l = 1, 2, ...q$

Los tipos de clustering más utilizados son la partición y la jerarquía completa de particiones (aquellas que contienen N particiones) Esta última jerarquía también puede definirse como un conjunto de $2N - 1$ clusters disjuntos o con inclusión unos en otros. Recientemente, también se ha trabajado con el concepto de $agrupamiento\ difuso$ ($fuzzy\ clustering$), en el cual las entidades pueden pertenecer a más de un cluster.

En el agrupamiento con restricciones, se añaden ciertos requerimientos a los clusters, como límites de peso o de cardinalidad, o conexión entre entidades (suponiendo que es conocida una matriz de adyacencia entre ellas).

2.1.3. Disimilitudes. Divergencia y distancia

Distancia entre dos individuos

Una vez realizada una adecuada selección de las variables a considerar, cada uno de los individuos sujetos al análisis nos vendrán representados por los valores que tomen estas variables en cada uno de ellos. Este es el punto de partida de la clasificación. Para clasificar adecuadamente los individuos deberemos determinar lo similares o disimilares (divergentes) que son entre sí, en función de lo diferentes que resulten ser sus representaciones en el espacio de las variables.

Para medir lo similares (o disimilares) que son los individuos existe una enorme cantidad de índices de similaridad y de disimilaridad o divergencia. Todos ellos tienen propiedades y utilidades distintas y habrá que ser consciente de ellas para su correcta aplicación al caso que nos ocupe.

En este trabajo nos centraremos en aquellos que indiquen la distancia entre dos individuos (considerando a los individuos como vectores en el espacio de las variables).

Las disimilitudes verifican las tres propiedades mencionadas en el paso c) de los citados para el Análisis Cluster:

$d_{kl} \geq 0$ para todo $k \neq l$ con $k, l = 1, ..., N$

$d_{kk} = 0$

$d_{kl} = d_{lk}$ para todo $k, l = 1, ..., N$

Además, por el hecho de ser distancias, verifican la desigualdad triangular:

$d_{kl} \leq d_{km} + d_{ml}$

Si además verificase la propiedad:

$d_{kl} \leq \text{máx}(d_{km}, d_{ml})$

conocida como desigualdad triangular ultramétrica, se dice que la distancia es ultramétrica.

Existe una gran cantidad de distancias y no existe una regla general para definir cuándo es conveniente utilizar una u otra. Dependiendo de cada caso concreto, la utilización de una u otra puede resultar más efectiva.

Distancia Euclídea:

La distancia euclídea es la medida de disimilaridad más conocida y más sencilla de comprender debido a que su definición coincide con el concepto más común de distancia. Dados dos individuos o entidades O_l, O_k de la muestra inicial O de los cuales se han medido p características en la matriz de datos B, la distancia euclídea entre ellos viene dada como sigue:

$$d_{lk} = d(O_l, O_k) = \|O_l - O_k\|_2 = \sqrt{\sum_{i=1}^{p}(O_{li} - O_{ki})^2}$$

donde O_{li}, O_{ki} denotan los elementos li, ki de la matriz de datos B

Pese a la sencillez de cálculo, tiene dos graves incovenientes:

-)Es sensible a las unidades de medida de las variables. Las diferencias entre los valores de variables medidas con valores altos contribuirán en mucha mayor medida que las diferencias entre los valores de las variables con valores bajos. Como consecuencia de ello, los cambios de escala determinan cambios en la distancia entre los individuos. Una posible vía de solución de este problema es la tipificación previa de las variables.

-)El segundo inconveniente no se deriva de forma directa de la utilización de este tipo de distancia, sino de la naturaleza de las variables. Si las variables utilizadas están correlacionadas, estas variables dan una información redundante (en gran medida, puesto que parte de las diferencias entre los valores individuales de algunas variables podrían explicarse por las diferencias en otras variables). Este efecto también es conocido como colinealidad. Como consecuencia de ello la distancia euclídea hace aumentar la disimilaridad entre individuos.

La solución a estos problema pasa por realizar un Análisis de Componentes Principales, que considere las variables principales (que estarán incorreladas) en lugar de las variables iniciales.

En conclusión, será recomendable utilizar distancia euclídea cuando las variables sean homogéneas y estén medidas en unidades similares, y/o cuando se desconozca la matriz de varianzas.

Distancia Manhattan o ciudad:

Análogamente, para las entidades O_k, O_l se define la distancia Manhattan o ciudad como:

$$d_{kl} = d(O_k, O_l) = \|O_k - O_l\|_1 = \sum_{i=1}^{p} |O_{li} - O_{ki}|$$

Esta distancia también presenta problemas con la colinealidad (redundancia de información).

Distancia al cuadrado de Mahalanobis:

Esta distancia tiene mucho prestigio en Estadística. Se trata de una distancia que tiene en cuenta no sólo las distancias que hay en cada una de las variables, sino que relativiza cada una de estas distancias respecto a la dispersión que tiene cada una de esas variables originales. Solventa los dos problemas expuestos de la distancia euclídea: Resulta invariante ante cambios de escala y por tanto, no depende de las unidades de medida. Además, corrige el efecto de redundancia. Supuesta conocida la matriz de varianzas Σ, se define esta distancia como:

$$d_{kl} = (O_k - O_l)'(\Sigma^{-1})(O_k - O_l)$$

Cuando las variables son incorreladas, esta distancia coincide con la llamada distancia euclídea normalizada.

Muchas otras distancias pueden ser consideradas dependiendo del caso particular a estudiar, pero estas en concreto resultan las más conocidas y utilizadas.

Distancia entre clusters

La aplicación del análisis cluster requiere no sólo el cálculo de las distancias entre los individuos iniciales, sino también la determinación de las distancias entre los grupos que se irán formando y/o entre un grupo y un individuo.

Esta necesidad de determinar las distancias entre grupos es especialmente importante en los métodos jerárquicos, que se tratarán más tarde.

Existen varias alternativas:

Distancia mínima: "$Nearest\ neighbour\ distance$"

Se puede definir la distancia entre un cluster y un individuo como la menor de las distancias entre los individuos del cluster y el individuo exterior considerado.

Si llamamos C_i al cluster formado por las entidades $(O_1, O_2, ..., O_i)$ y O_j a la entidad exterior, se define la distancia entre C_i y O_j como:

$$d(C_i, O_j) = \min_{O_l \in C_i} d(O_l, O_j)$$

Análogamente, siguiendo este criterio, puede definirse la distancia entre dos clusters C_i y C_j como la mínima de las distancias entre un individuo de C_i y otro de C_j:

$$d(C_i, C_j) = \min_{O_l \in C_i, O_k \in C_j} d(O_l, O_k)$$

Como veremos, la distancia mínima será la utilizada en el algoritmo jerárquico de clasificación conocido como método de la distancia mínima o single linkage (método de enlace simple, o de vecino más próximo).

Este método será espacio-contractivo (tiende a aproximar los individuos más de lo que indicarían sus disimilitudes o distancias iniciales). Ha sido reivindicado "matemáticamente preferible"por sus propiedades por algunos autores. A su vez, ha sido muy criticado por ser muy sensible en aquellos casos en los que existen individuos perturbadores entre clusters bien diferenciados (los llamados casos con ruido).

Distancia máxima: "$Further\ neighbour\ distance$"

Otra forma de definir la distancia entre un cluster y un individuo exterior es considerar el máximo de las distancias entre O_j y los individuos del cluster C_i:

$$d(C_i, O_j) = \max_{O_l \in C_i} d(O_l, O_j)$$

y la distancia entre dos clusters, análogamente se define como:

$$d(C_i, C_j) = \max_{O_l \in C_i, O_k \in C_j} d(O_l, O_k)$$

Este distancia es utilizada para la obtención de la clasificación jerárquica ascendente, considerando la distancia entre clusters como la distancia entre los individuos más alejados. Este método recibe el nombre de complete linkage (método

de enlace completo o del vecino más lejano). Por modificar la distancia en sentido inverso que la propuesta anterior, este método es espacio-dilatante (tiende a separar a los individuos en mayor medida que la indicada por sus disimilitudes iniciales). Es menos utilizado y se encuentran en decadencia, ya que presenta inconvenientes como alargar mucho el proceso y dar como resultado clusterings encadenados.

Mientras el método de la distancia mínima asegura que la distancia entre los individuos más próximos de un cluster es siempre menor que la distancia entre elementos de distintos clusters, el de la distancia máxima va a asegurar que la distancia máxima dentro de un cluster será menor que la distancia entre cualquiera de sus elementos y los elementos más alejados de los demás clusters.

Distancia entre centroides

También se puede definir la distancia entre el cluster C_i y el individuo O_j como la distancia entre el centro de gravedad o centroide de C_i y O_j

Si $\bar{i}$ es el centroide de C_i, calculado mediante:

$$\bar{i} = \frac{1}{|C_i|} \sum_{O_k \in C_i} O_k$$

entonces se tendrá:

$$d(C_i, O_j) = d(\bar{i}, O_j)$$

siendo la distancia entre dos clusters C_i y C_j la distancia entre sus centroides:

$$d(C_i, C_j) = d(\bar{i}, \bar{j})$$

donde $\bar{j}$ denota el centroide de C_j

Aplicar un método para el Análisis Cluster basado en esta distancia lo hará también espacio-conservativo, pero presenta el inconveniente de dejarse influir excesivamente por los grupos de mayor tamaño.

Análogamente a la distancia entre individuos, existen muchas otras posibles a considerar en dependencia de cada caso particular, del objetivo a conseguir y de los inconvenientes de cada distancia.

2.1.4. Criterios

Como se adelantó en el capítulo anterior, la elección de un criterio determina de forma sustancial la solución que se obtendrá. Al no existir solución única para este problema, puede ser interesante resolver el mismo problema para distintos criterios, y analizar en una etapa final las diferencias entre los agrupamientos obtenidos mediante un criterio u otro. Consideramos criterios basados en la disimilitud entre entidades para expresar la separación o la homogeneidad para un cluster C_j.

Separación: La separación de C_j puede ser medida mediante:

1) El $split$ $s(C_j)$ de C_j, o mínima disimilitud entre una entidad de C_j y otra fuera de C_j:

$$s(C_j) = \min_{k/O_k \in C_j, l/O_l \notin C_j} d_{kl}$$

2) El $corte$ $c(C_j)$ de C_j, o suma de las disimilitudes entre entidades de C_j y entidades fuera de C_j:

$$c(C_j) = \sum_{k/O_k \in C_j} \sum_{l/O_l \notin C_j} d_{kl}$$

Se podría considerar tambien el $corte\ normalizado$, que corrige la medida previa eliminando el efecto del tamaño del cluster. Para ello, basta dividir $c(C_j)$ por $|C_j|(N - |C_j|)$

Homogeneidad: La homogeneidad de C_j puede medirse mediante:
1) El $diámetro$ C_j de C_j, máxima disimilitud entre entidades de C_j:

$$d(C_j) = \max_{k,l/O_k, Ol \in C_j} d_{kl}$$

2) El $radio$ $r(C_j)$ de C_j, mínimo para todas las entidades O_k de C_j de la máxima disimilitud entre O_k y otra entidad de C_j:

$$r(C_j) = \min_{k/O_k \in C_j} \max_{l/O_l \in C_j} d_{kl}$$

3) La $estrella\ st(C_j)$ de C_j, mínimo para todas las entidades O_k de C_j de la suma de disimilitudes entre O_k y las otras entidades de C_j:

$$st(C_j) = \min_{k/O_k \in C_j} \sum_{l/O_l \in C_j} d_{kl}$$

4) El $clique\ cl(C_j)$ de C_j, suma de las disimilitudes entre entidades de C_j:

$$cl(C_j) = \sum_{k,l/O_k,O_l \in C_j} d_{kl}$$

Podría considerarse también la $estrella\ normalizada$ y el $clique\ normalizado$ definidos como $st(C_j)$ dividido por $|C_j| - 1$ y $cl(C_j)$ dividido por $|C_j|(|C_j| - 1)$ respectivamente.

Si las entidades O_j son puntos x de un espacio euclídeo de dimensión p, hay más conceptos que son útiles. La homogeneidad de C_j se mide respecto a un centro de C_j (un punto de C_j), como en las definiciones de $r(C_j)$ y $st(C_j)$. Se puede utilizar:

1) La $suma\ de\ cuadrados\ ss(C_j)$ de C_j, suma de los cuadrados de las distancias euclídeas entre entidades de C_j y su centroide $\overline{x}$:

$$ss(C_j) = \sum_{k/O_k \in C_j} (\|x_k - \overline{x}\|_2)^2$$

donde $\|\cdot\|_2$ denota la distancia euclídea y

$$\overline{x} = \frac{1}{|C_j|} \sum_{k/O_k \in C_j} x_k$$

2) La $varianza\ v(C_j)$ de C_j definida como $ss(C_j)$ dividida por $|C_j|$

3) El $radio\ continuo\ cr(C_j)$ de C_j definido como:

$$cr(C_j) = \min_{x \in \mathbb{R}^p} \max_{k/O_k \in C_j} \|x_k - x\|_2$$

4) La $estrella\ continua\ cts(C_j)$ de C_j definida como:

$$cts(C_j) = \min_{x \in \mathbb{R}^p} \sum_{k/O_k \in C_j} \|x_k - x\|_2$$

Los conceptos definidos anteriormente dan lugar a dos familias de criterios, con el objetivo de ser maximizados para la separación y minimizados para la homogeneidad. Corresponden a centrarse en el peor de los clusters o considerar todos los clusters (o valores medios) respectivamente.

Considerando particiones P de O en M clusters, el $split\ \ s(P_M)$ de la partición P_M es el menor split de sus clusters, el $diámetro\ \ d(P_M)$ es el mayor diámetro de sus clusters, y así con todos los criterios. El $split\ medio\ \ av(P_M)$ de P_M es la suma de los splits de sus clusters dividida por M, el $diámetro\ medio\ \ ad(P_M)$ de P_M es la suma de los diámetros de sus clusters dividida por M, y así con todos de nuevo.

Definiciones similares pueden darse para el empaquetamiento, el cubrimiento y la jerarquía, vistos como conjuntos de $2N - 1$ clusters.

De nuevo, es necesario realizar algunas observaciones sobre esto:

- No todos los criterios son independientes. Por ejemplo, minimizar el $clique\ medio$ es equivalente a maximizar el $corte\ medio$.
- En segundo lugar, hay criterios que expresan simultáneamente la separación y la homogeneidad, como es el caso de minimizar la suma de cuadrados dentro de los clusters (criterio de homogeneidad); equivalente a maximizar la suma de cuadrados entre los clusters (criterio de separación).
- Criterios como $r(C_j)$, $st(C_j)$, $ss(C_j)$ y $v(C_j)$ hacen uso de un centro del cluster. Este centro puede ser considerado como representante del cluster en diversas aplicaciones.
- En última instancia, los criterios definidos para las particiones pueden ser usados de diversas formas. Pueden ser optimizados globalmente (exacta o aproximádamente) en partición; o localmente en agrupación jerárquica, donde los cambios de una partición a la siguiente están sujetos a restricciones.

2.1.5. Algoritmos de clustering

Según el objetivo a lograr o la información de que se disponga, puede ser útil considerar algún tipo de algoritmo en particular. La forma de proceder de cada uno de ellos es distinta, logrando diferentes resultados en el proceso de agrupación. Una clasificación general de los algoritmos se puede hacer como sigue:

Aglomerativo o divisivo:

Un algoritmo será aglomerativo o ascendente si se parte inicialmente de todos los individuos, que se van progresivamente fusionando formando grupos que constituyen las sucesivas particiones. Por el contrario, será divisivo o descendente si se parte de todo el conjunto de individuos como un conglomerado y se va sucesivamente subdividiendo en grupos más pequeños. En cada etapa de ascenso/descenso los individuos quedan agrupados en nuevos clusters, lo cual hace interesante conocer los clusters formados a cada nivel del procedimiento.

Jerárquico o no jerárquico:

En una clasificación no jerárquica se forman grupos homogéneos sin establecer relaciones entre ellos. En una clasificación jerárquica, los grupos se van fusionando o subdividiendo sucesivamente siguiendo una prelación o jerarquía, decreciendo la homogeneidad conforme se van haciendo más amplios.

En el caso jerárquico, la representación de dicha jerarquía de clusters obtenida suele llevarse a cabo por medio de un diagrama en forma de árbol invertido llamado dendrograma, en el que las sucesivas fusiones de las ramas a los distintos niveles nos informan de las sucesivas fusiones de los grupos en grupos de superior nivel (en general, a mayor tamaño se tiene menor homogeneidad).

2.2. Agrupación Jerárquica

En los métodos jerárquicos, los individuos no se particionan en clusters de una sola vez sino que se van haciendo particiones sucesivas a distintos niveles de agregación o agrupamiento.

Los métodos jerárquicos suelen subdividirse en métodos aglomerativos (ascendentes), que van sucesivamente fusionando grupos en cada paso; y métodos divisivos (descendentes), que van desglosando en clusters cada vez más pequeños el conjunto total de datos.

2.2.1. Algoritmos de agrupación jerárquica aglomerativa

Estos son los métodos más usados en el Análisis Cluster. Se inicia con una partición inicial en N clusters de una entidad cada uno (cada O_i comienza siendo un cluster en sí) y mediante fusiones sucesivas, todas las entidades pertenecen al mismo cluster. Considerar los distintos niveles del proceso lleva a conocer los clusters en que se van agrupando los individuos.

Inicialización
$P_N = \{C_1, ..., C_N\}$;
$C_j = \{O_j\}$, $j = 1, 2, ..., N$;
$k = 1$;

Paso actual:
While $N - k > 1$ hacer:
Seleccionar $C_i, C_j \in P_{N-k+1}$ siguiendo un criterio local;
$C_{N+k} = C_i \bigcup C_j$;
$P_{N-k} = (P_{N-k+1} \bigcup \{C_{N+k}\}) \setminus \{C_i, C_j\}$;
$k = k + 1$;
Acabar While

Un criterio local es aquel que usa sólamente la información dada en D y en la partición actual. De esta forma, el algoritmo no tiene en cuenta cómo se ha obtenido la partición actual para obtener las siguientes. Este es el caso del método

de enlace simple, el cual fusiona en cada etapa los dos clusters para los cuales la menor disimilitud intercluster es mínima.

Hay resultados de la teoría de grafos que pueden reformularse para este caso: Sea $G = (V, E)$ un grafo completo cuyos vértices v_k están asociados con las entidades O_k, para $k = 1, ..., N$ y cuyas aristas $\{v_k, v_l\}$ están ponderadas mediante las disimilitudes d_{kl}. Sea MST un árbol de mínima expansión de G con respecto a las disimilitudes d_{kl} (es decir, un subgrafo conexo de G en el que cualesquiera dos vértices están conectados por un camino de forma que se cubren todos ellos sin contener ningún ciclo, realizado respecto a la ponderación de las aristas con las disimilitudes d_{kl} obteniendo la menor disimilitud total posible).

Proposición 2.1. *(Rosenstiehl, L'arbre minimum d'un graphe, 1967 [4]) Los valores del split para todo subconjunto de entidades de O, y por tanto, para todas las particiones de O, pertenecen al conjunto de valores de disimilitud asociados a las aristas de MST.*

Corolario 2.2. *(Delattre y Hansen, Bicriterion cluster analysis, 1980 [5]) El algoritmo de enlace simple proporciona particiones de máximo split a todos los niveles de la jerarquía.*

Demostración: Sea HI el conjunto de todas las jerarquías de las particiones de O. Entonces, el resultado anterior consiste en probar que para toda $H \in HI$ el algoritmo de enlace simple mazimiza el split de todos los clusters existentes en cada nivel de la jerarquía.

Sea H^* una jerarquía de particiones de O con máximo split, y sea d_{kl} la mínima disimilitud entre pares de entidades de O. Puede ocurrir que $\{O_k, O_l\}$ sea un cluster de H^* o que no lo sea. En caso de que no lo fuese, se define una jerarquía modificada $\hat{H}$ como sigue:

a. Para todos los clusters C_i pertenecientes a H^* tales que $O_k \in C_i$ y $O_l \notin C_i$, se añade O_l a C_i.

b. Para todos los clusters C_j pertenecientes a H^* tales que $O_l \in C_j$ y $O_k \notin C_j$, se elimina O_l de C_j.

c. Se ha formado una clase duplicada: en a. si O_l se unió antes a un cluster C_p que contenía a O_k; y en b. si O_l se unió antes a un cluster C_q que no contenía a O_k. Este cluster duplicado se corresponde a $C_p \bigcup \{O_l\}$ en el primer caso, y a $C_q \setminus \{O_l\}$ en el segundo.

d. Añadir el cluster $\{O_k, O_l\}$

Todos los clusters que hayan sido modificados contenían o bien a O_k o bien a O_l, pero en ningún caso a ambos. Por tanto, su split era menor o igual a d_{kl}. Ahora, han sido reemplazados por clusters que contienen a $\{O_k, O_l\}$, cuyo split no puede ser menor. En consecuencia, $s(\hat{H}) \geq s(H^*)$.

Entonces, en H^* o en $\hat{H}$ se tiene el cluster $\{O_k, O_l\}$. En ese caso, las entidades O_k, O_l pueden ser fusionadas como en el algoritmo de enlace simple y el argumento anterior puede repetirse. Con esto concluye la prueba. □

Para otros criterios, las particiones obtenidas tras varios pasos de un algoritmo aglomerativo no son necesariamente óptimas. Por ejemplo, el método de enlace completo fusiona en cada paso los dos clusters para los cuales el cluster obtenido (así como la partición resultante) tiene menor diámetro. Tras dos o más pasos, la partición obtenida no tiene necesariamente el mínimo diámetro.

Una fórmula paramétrica da nuevos valores de disimilitud entre los clusters C_k y C_i, C_j cuando estos dos últimos se fusionan. Este método del cálculo de distancias es conocido como Método flexible de Lance y Williams:

$$d_{k,i\bigcup j} = \alpha_i d_{ik} + \alpha_j d_{jk} + \beta d_{ij} + \delta |d_{ik} - d_{jk}|$$

Donde los parámetros $\alpha_i, \alpha_j, \beta$ y δ adquieren un determinado valor según la importancia de la disimilitud a la que acompañan en el problema, y los clusters que se fusionan en cada iteración son aquellos que tienen la menor disimilitud. Se puede obtener una implementación uniforme del algoritmo de agrupación jerárquica de O($N^2 logN$).

Se puede llegar a mejores resultados en algunos casos, como encontrando el árbol de mínima expansión MST de G, ordenando sus aristas según sus valores en orden no decreciente y fusionando entidades de aristas sucesivas. En este caso se llega a una implementación del algoritmo de enlace simple de $\theta(N^2)$.

En cada iteración, los clusters se corresponden a componentes conectadas de un grafo con el mismo conjunto de vértices que G y el mismo conjunto de aristas del MST considerado.

La siguiente propiedad de reducibilidad

$$d(C_i, C_j) \leq \text{mín}\{d(C_i, C_k), d(C_j, C_k)\}$$

implica lo siguiente:

$$\text{mín}\{d(C_i, C_k), d(C_j, C_k)\} \leq d(C_i \bigcup C_j, C_k) \quad \forall i, j, k;$$

Esto quiere decir que, unir dos clusters C_i, C_j menos disimilares entre sí que con otro cluster C_k no puede hacer la disimilitud resultante con C_k menor que la menor de las iniciales. Para este algoritmo, las disimilitudes $D = (d_{kl})$ inducen una relación de vecindad cercana entre uno o más pares de vecinos cercanos. Cuando se verifica la propiedad de reducibilidad, cada par de vecindades cercanas se unirán entre ellas antes que hacerlo con otro cluster. Actualizando cadenas de vecindades cercanas se consigue un algoritmo $\theta(N^2)$ de agrupación jerárquica aglomerativa para el criterio de la varianza (o varianza media). Este resultado se extiende al método de enlace simple, al método de enlace completo y al método de la media. En el caso en que las entidades de O pertenecen a un espacio euclídeo de baja dimensión y las disimilitudes miden la distancia entre los puntos que representan, se puede recurrir a técnicas de geometría computacional, obteniendo algoritmos incluso más rápidos.

2.2.2. Algoritmos de agrupación jerárquica divisiva

Estos algoritmos son menos usados que los anteriores. Parten de un cluster inicial que contiene a todas las entidades, y mediante sucesivas biparticiones de un cluster cada vez se logra que todas las entidades pertenezcan a diferentes clusters.

Inicialización
$P_1 = \{C_1\} = \Big\{\{O_1, ..., O_N\}\Big\}$;
$k = 1$;

Paso actual:
While $k < N$
Seleccionar $C_j \in P_k$ siguiendo un primer criterio local;
Particionar C_j en C_{2k} y C_{2k+1} siguiendo un segundo criterio local;
$P_{k+1} = \Big(P_k \bigcup \{C_{2k}\} \bigcup \{C_{2k+1}\}\Big) \setminus \{C_j\}$;
$k = k + 1$
Acabar While

El primer criterio local no es crucial, sólo determina el orden en que los clusters se biparticionan. La verdadera dificultad se encuentra en biparticionar el cluster elegido de acuerdo al segundo criterio, pues requiere algoritmos específicos para cada caso concreto pudiendo llegar a ser NP-duros. En consecuencia, se proponen pocos algoritmos divisivos.

Para el criterio del mínimo diámetro, se explota una propiedad de cualquier árbol de máxima expansión del grafo G con respecto a las disimilitudes d_{kl} definido anteriormente (un subgrafo conexo de G en el que cualesquiera dos vértices están conectados por un camino de forma que se cubren todos ellos sin contener ningún ciclo, realizado respecto a la ponderación de las aristas con las disimilitudes d_{kl} obteniendo la mayor disimilitud total posible). Consideremos una bipartición $\{C_1, C_2\}$ de O con diámetros $d_1 = d(C_1)$ y $d_2 = d(C_2)$, y supongamos sin pérdida de generalidad $d_1 \geq d_2$. Sea MST' el árbol de máxima expansión de G respecto a las disimilitudes d_{kl} y C_{pq} un ciclo formado por una arista $\{v_p, v_q\} \notin MST'$ y el único camino uniendo v_p con v_q en MST'.

Teorema 2.3. *(Hansen y Jaumard, Minimum sum of diameters clustering, 1987 [10]) Para cualquier bipartición $\{C_1, C_2\}$ el diámetro d_1 es igual a la mayor disimilitud d_{pq} asociada a la arista $\{v_p, v_q\} \notin MST'$ tal que C_{pq} es impar, o a d_{kl} para alguna arista $\{v_k, v_l\} \in MST'$ con $d_{kl} \geq d_{pq}$.*

Demostración: Se aplica el algoritmo de Kruskal a G para obtener el MST'. Este algoritmo busca un subconjunto de aristas que incluya todos los vértices formando un árbol y donde el valor total de todas las aristas del árbol es mínimo. Se consideran las aristas en orden decreciente de sus pesos, y se mantienen si y sólo si no cierran un ciclo.

Sea $\{v_p, v_q\}$ la primera arista que cierra un ciclo impar, denotado por C_{pq}. Veamos que $d_1 \geq d_{pq}$.

Por construcción de MST', $d_{kl} \geq d_{pq}$ para todas las aristas $\{v_k, v_l\}$ de C_{pq} que sean diferentes de $\{v_p, v_q\}$. Entonces, si $d_1 < d_{pq}$ cualquiera dos entidades tales que sus vértices asociados sean adyacentes en C_{pq} no deberían pertenecer al mismo cluster. De esta forma, entidades asociadas a vértices en C_{pq} deberían pertenecer alternativamente a C_1 y C_2, lo cual es imposible debido a que C_{pq} es impar.

Veamos qué ocurre para aristas $\{v_i, v_j\}$ con $d_{ij} > d_{pq}$ que cierran un ciclo C_{ij}. De nuevo por construcción de MST', $d_{kl} \geq d_{ij}$ para todas las aristas en C_{ij} distintas de $\{v_i, v_j\}$. Si $d_1 = d_{ij}$, tanto O_i como O_j pertenecen a C_1. Pero entonces, al menos otras dos entidades O_k y O_l asociadas con vértices adyacentes de C_{ij}

deben pertenecer al mismo cluster. Entonces, $d_{kl} > d_{ij}$ entraría en contradicción con $d_1 = d_{ij}$. Esto quiere decir que debe alcanzarse la igualdad $d_{kl} = d_{ij}$. □

Aplicando este teorema, se puede probar de forma sencilla el siguiente resultado:

Proposición 2.4. *(Guénoche, Partitions with minimum diameter, 1989 [6]; Monma y Suri, Partitioning points and graphs, 1991 [7]) La única bicoloración de MST' define una bipartición de mínimo diámetro de O.*

Demostración: Un árbol de máximo spanning admite una bicoloración única de sus vértices salvo la permutación de los colores. El diámetro de la bipartición $\{C_1, C_2\}$ corresponde a la mayor disimilitud de una arista con ambos extremos del mismo color, es decir, una arista que no pertenezca a MST' y que cierre un ciclo impar con un camino de MST'. Por tanto, es igual a d_{pq} que por el teorema anterior es mínimo. □

Usando esta proposición se consigue un algoritmo jerárquico divisivo O(N^3) a todos los niveles. Mediante una implementación más cuidadosa, construyendo simultáneamente árboles de máxima expansión de G respecto a las disimilitudes a todos los niveles, se puede mejorar el tiempo a O($N^2 logN$).

De aquí, se sigue que hay al menos O(N) valores candidatos a diámetro de una bipartición. Esta propiedad puede ser usada en algoritmos divisivos para la agrupación jerárquica con el criterio del diámetro medio. La existencia de una bipartición con diámetros dados se prueba resolviendo una ecuación cuadrática boleana o mediante algoritmos especializados en etiquetado. El algoritmo resultante toma un tiempo O($N^3 logN$). Resulta más difícil construir un algoritmo para el clustering jerárquico divisivo con método de la media (biparticionar O para maximizar la disimilitud media entre distintos clusters es fuertemente NP-duro). Sin embargo, se pueden abordar problemas con tamaño moderado ($N \leq 40$) usando programación hiperbólica y programación cuadrática 0-1. Para varios criterios, cuando las entidades son puntos de $\mathbb{R}^2$ hay hiperplanos separando los clusters. Esta propiedad es explotada en un algoritmo para el clustering jerárquico divisivo con criterio mínima suma de cuadrados en espacios de baja dimensión, resolviendo casos con $N \leq 20000$ en $\mathbb{R}^2$, $N \leq 500$ en $\mathbb{R}^3$ y $N \leq 150$ en $\mathbb{R}^4$

2.2.3. Criterios globales

Anteriormente ya se afirmó que una jerarquía completa de las particiones puede ser vista como un conjunto de $2N-1$ clusters. Optimizar una función objetivo definida sobre este conjunto de clusters aún no ha sido explorado salvo para el criterio del split medio (donde el split de O consigo mismo se asume 0). El algoritmo de enlace simple maximiza este valor. Los resultados de la agrupación jerárquica pueden ser representados gráficamente en un dendrograma o espalier. Las líneas verticales corresponden a entidades o clusters y las horizontales unen las líneas verticales representando la unión de clusters. La altura de las lineas horizontales corresponden al valor de la disimilitud actualizada entre los clusters que se han unido. Sirve como medida de homogeneidad o separación de los clusters obtenidos. En los espaliers, la longitud de las líneas horizontales es utilizada como segunda medida de homogeneidad o separación de los clusters. Cuando se alcanza la condición de reducibilidad, las disimilitudes actualizadas verifican la inecuación ultramétrica:

$$d'_{kl} \leq \max(d'_{kj}, d'_{jl}) \quad \forall j, k, l$$

Es decir, un algoritmo de agrupación jerárquica transforma una disimilitud inicial $D = (d_{kl})$ en disimilitudes ultramétricas $D' = (d'_{kl})$. Esto sugiere más criterios, como minimizar:

$$\sum_{k,l}(d_{kl} - d'_{kl})^2 \quad \text{ó} \quad \sum_{k,l}|d_{kl} - d'_{kl}|$$

En el primer caso, que es NP-duro, una combinación del algoritmo de la media con branch-and-bound resuelve casos con $N \leq 20$. En el segundo, mediante branch-and-bound se resuelven casos ligeramente mayores. Los métodos heurísticos usan métodos de penalización en los cuales no cumplir la inecuación ultramética es penalizado. También pueden considerarse proyecciones iteradas, sin olvidar que al realizar una proyección se puede perder información relevante.

2.3. Particionamiento

En el caso de proceder al clustering por partición y no por jerarquía, se consideran otros métodos para la obtención de los clusters. Se exponen a continuación algunos de los posibles a utilizar.

2.3.1. Programación Dinámica

En los problemas de clustering en una dimensión, las entidades $O_1, ..., O_N$ se corresponden a puntos $x_1, ..., x_N$ de una línea euclídea. En estos casos, el problema de clustering se resuelve mejor mediante programación dinámica. En general, funciona bien cuando los clusters tienen la llamada $propiedad\ de\ cadena$ (son puntos consecutivos en la línea). Supongamos que las entidades $O_1, ..., O_N$ se encuentran en orden de valores no decrecientes $x_1, ..., x_N$. Sea $f(C_j)$ la contribución del cluster C_j a la función objetivo (supuesta aditiva en los clusters y con objeto de ser minimizada) y F_m^l el valor óptimo de una agrupación de $O_1, ..., O_m$ en l clusters. La ecuación de recurrencia puede ser escrita como sigue:

$$F_m^l = \min_{\{k \in l, l+1, ..., m\}} \left\{ F_{k-1}^{l-1} + f(C_m) \right\}$$

donde $C_m = \{O_k, O_{k+1}, ..., O_m\}$.

Es posible lograr un algoritmo $O(N^2)$ para varios criterios mediante actualización para computar los valores de $f(C_j)$ para todos los clusters potenciales.

Hay que tener en cuenta que la propiedad en cadena no siempre se alcanza (por ejemplo, los clusters óptimos para particionamiento en clique en casos unidimensionales no tienen por qué satisfacerla). En su lugar, se tiene una propiedad de anidamiento más débil: Sea $[C_j]$ el rango de las entidades $O_k, ..., O_l$ de C_j, es decir, $[x_k, x_l]$. Entonces, para cualesquiera dos clusters C_i, C_j en el conjunto de particiones óptimas se tiene:

$$[C_i] \bigcap [C_j] = \emptyset \quad \text{ó} \quad [C_i] \subseteq [C_j] \quad \text{ó} \quad [C_j] \subseteq [C_i]$$

Así, rangos de cualesquiera dos clusters son o bien disjuntos o están incluidos uno dentro del otro. Explotando esta propiedad se llega a un algoritmo en tiempo polinomial para el particionamiento en clique en una dimensión (basado en programación dinámica).

Cuando se procede en espacios de más alta dimensión, no parece haber una propiedad en cadena equivalente al caso unidimensional. En algunos casos la

ecuación de recurrencia puede ser extendida. Varios autores han propuesto imponer un orden en las entidades (por ejemplo, el orden de puntos en una curva de Peano o el orden de un recorrido en un viaje del problema del viajero) y aplicar la programación dinámica al problema unidimensional resultante. Este procedimiento obtiene rápidamente una solución óptima para una aproximación del problema dado. Su proximidad al óptimo del problema en sí depende del primer paso, que en parte es arbitrario.

Para obtener una solución óptima en el caso general, se debe utilizar programación dinámica no seriada. Denotando como F_S^l al valor óptimo del clustering de las entidades del subconjunto S en l clusters, la ecuación de recurrencia se convierte en:

$$F_S^l = \min_{C_m \subset S} \{F_{S \setminus C_m}^{l-1} + f(C_m)\}$$

Aplicar esta ecuación toma un tiempo exponencial en N, de modo que sólo pequeños conjuntos de entidades ($N \leq 20$) pueden ser considerados. Añadiendo restricciones se acelera la computación (los clusters deben ser pequeños).

2.3.2. Algoritmos basados en teoría de grafos

Se mencionó anteriormente que el algoritmo de enlace simple proporciona particiones óptimas para el criterio del split a todos los niveles de la jerarquía (corolario 2.2). Por tanto, también es un algoritmo para maximizar el split de una partición de O en M clusters. El problema de maximizar el split medio o la suma de splits de particiones como éstas están relacionados pero son diferentes. La solución se basa en el siguiente resultado:

Proposición 2.5. *(Hansen, Maximum sum-of-splits clustering, 1989 [8]) Sea* $C = \{C_1, C_2, ..., C_{2N-1}\}$ *el conjunto de clusters obtenidos tras aplicar el algoritmo de enlace simple a* O*. Para cualquier MST y para cada* $M = 2, 3, ..., N-1$ *existe una partición* P_M *de* O *tal que todos sus clusters pertenecen al conjunto de* $2N-1$ *clusters obtenido de aplicar el algoritmo de enlace simple al MST.*

Demostración: En primer lugar, veamos que para un MST dado y para cualquier M existe una partición con máxima suma de split P_M tal que $C_1, ..., C_M$ inducen subgrafos conexos de MST.

Consideremos una partición P_M donde no se tenga esta propiedad. Cada cluster C_i induce uno o varios subrafos conexos de MST. Sea $p > M$ el número de tales subgrafos inducidos por todos los clusters de P_M. Sea C_k el cluster

que induce varios subgrafos conexos y C_k^1 un subgrado conexo inducido tal que $s(C_k^1) = s(C_k)$, como se muestra en la siguiente figura:

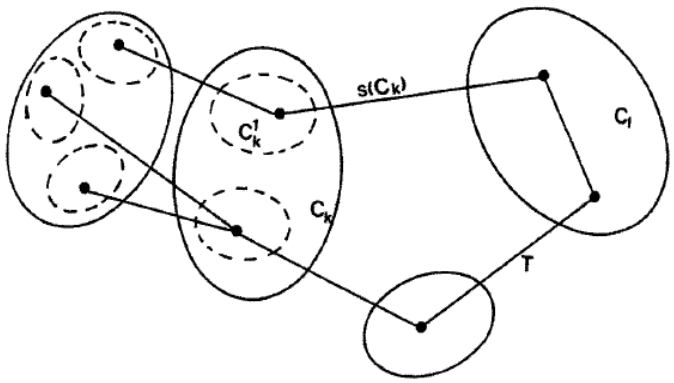

Figura 1: Subgrafos conexos de MST inducidos por una partición P_M.

Sea C_l el cluster que incluye el vértice externo a C_k en la arista que define $s(C_k)$. Nótese que $s(C_l) \leq s(C_k)$. Considérese entonces la partición obtenida de P_M reemplazando C_k por $C_k \setminus C_k^1$ y C_l por $C_l \bigcup C_k^1$. Entonces, $s(C_k \setminus C_k^1) \geq s(C_k)$ y $s(C_l \bigcup C_k^1) \geq s(C_l)$, por lo que la suma de splits no puede decrecer. Si aumenta, la partición P_M no es óptima. Si se mantiene, nótese que p ha disminuido en una unidad, por lo que el resultado se consigue por inducción.

Veamos a continuación que el valor óptimo es independiente del MST elegido cuando éste no es único: Supongamos que MST_1 y MST_2 son árboles de mínimo spanning. Una partición P_M^1 de máxima suma de splits para MST_1 puede ser convertida en una partición P_M^2 con una suma de splits no menor a la original para MST_2 utilizando el razonamiento anterior. Como el análogo comenzando en MST_2 también es cierto, las particiones de máxima suma de splits P_M^1 y P_M^2 deben tener igual valor. Esto lleva a afirmar que el óptimo es independiente del MST elegido.

Ahora, fijado un MST, veamos que existe una partición $P_M = \{C_1, ..., C_M\}$ de máxima suma de splits tal que todos sus clusters pertenecen a la jerarquía H proporcionada por el algoritmo de enlace simple. Un cluster C_k pertenece a H si y sólo si su split $s(C_k)$ verifica $d_{rs} \leq s(C_k)$ para toda arista $\{v_r, v_s\}$ de MST tal que v_r y v_s pertenecen a C_k. Se aplica inducción en el número de aristas q que no verifican esta condición. Considérese una partición P_M tal que no se satisface

la condición, es decir, existe un cluster C_k tal que $d_{rs} > s(C_k)$ para una arista $\{v_r, v_s\}$ de MST tal que v_r y v_s pertenecen a C_k:

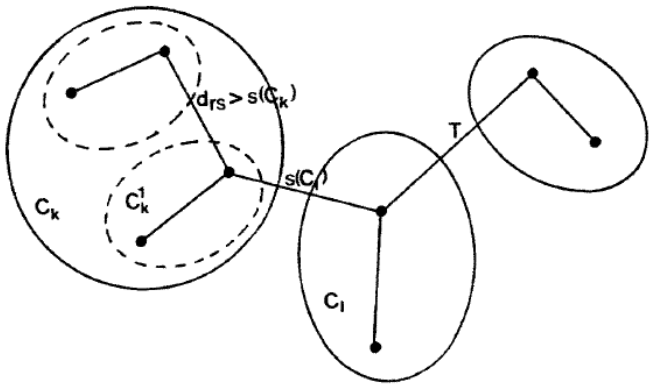

Figura 2: Clusters de H y de P_M.

Sea MST_k el subárbol de MST inducido por los vértices de C_k y C_k^1 el cluster asociado con los vértices de un subárbol de MST_k obtenido al eliminar $\{v_r, v_s\}$ y tal que $s(C_k^1) = s(C_k)$. Sea C_l de nuevo el cluster que incluye el vértice externo a C_k de la arista que define $s(C_k)$. Considérese de nuevo la partición obtenida de P_M reemplazando C_k por $C_k \setminus C_k^1$. De nuevo, también se tiene $s(C_k \setminus C_k^1) \geq s(C_k)$, $s(C_l \bigcup C_k^1) \geq s(C_l)$ y la suma de splits no puede decrecer. Si aumenta, la partición $\{C_1, ..., C_M\}$ no es óptima. Si se mantiene, entonces q ha disminuido en una unidad. Basta con continuar mediante inducción para obtener el resultado. □

Considérese entonces el grafo dual del dendrograma correspondiente al algoritmo de enlace simple (como se define en el anterior trabajo de Hansen). Puede verse que cualquier partición de O en M clusters de C corresponde a un camino con M arcos en ese grafo.

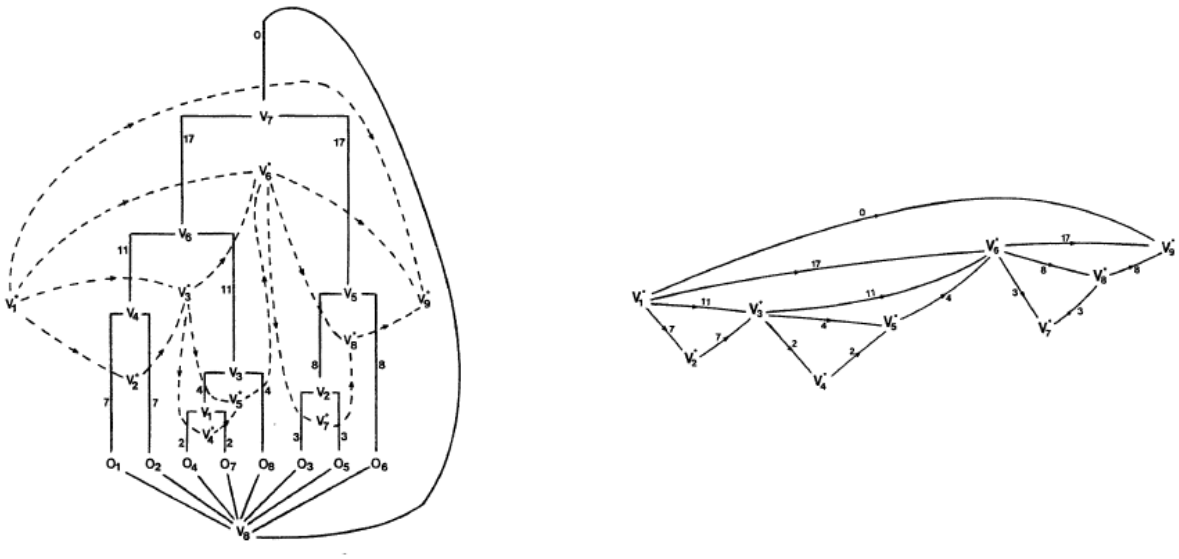

Figura 3: Dendograma de un grafo y su dual a la izquierda. Ponderación del grafo dual a la derecha.

Después, ponderando los arcos del grafo dual por el split de los clusters asociados con las aristas del dendrograma que cruzan y usando programación dinámica para encontrar una restricción a la cardinalidad del camino más largo se logra una partición P_M^* con máximo split medio en un tiempo $\theta(N^2)$.

La relación entre la coloración de grafos y encontrar una bipartición con mínimo diámetro fue mencionada en secciones anteriores. Se puede extender al caso general:

Consideremos la secuencia de grafos parciales $G_t = (G, E_t)$ donde $E_t = \{\{v_k, v_l\}/d_{kl} \geq t\}$ para valores de t iguales a las disimilitudes entre pares de entidades de O ordenadas en orden decreciente. Cada uno de estos grafos parciales G_t es colorable óptimamente. Es decir, colorable con $\gamma(G_t)$ colores, donde $\gamma(G)$ es el menor p tal que G admite una p-coloración. Entonces:

Proposición 2.6. *(Christofides, Graph Theory. An Algoritmic Approach, 1975 [9]; Hansen y Jaumard, Minimum sum of diameters clustering, 1987 [10]) Sea t la menor disimilitud tal que el grafo parcial $G_t = (V, E_t)$ de G es M-colorable. Entonces, las clases de color en cualquier coloración óptima de G_t definen una partición de mínimo diámetro de O en M clusters.*

Demostración: Si G_t denota el último grafo parcial M-colorable, sea $G_{t'}$ el siguiente grafo parcial en la secuencia y P^*_M la partición de O en M clusters inducida por las clases de color de una coloración óptima de G_t. Para cualquier O_l y $O_k \in O$ tal que $d_{kl} \geq t$, O_l y O_k son adyacentes en G_t. Por tanto $d(P^*_M) < t$, es decir, $d(P^*_M) \leq t'$. Como $G_{t'}$ no puede ser colorable en M colores, cualquier partición de G en M subconjuntos debe contener dos vértices O_p y O_q en la misma clase y ser adyacentes en $G_{t'}$, es decir, con $d_{pq} \geq t'$. Esto implica que no existe una partición P'_M de O con $d(P'_M) < t'$ y P^*_M tiene mínimo diámetro en el conjunto de todas las posibles particiones de O en M clusters. □

Esta relación se puede usar en sentido contrario para mostrar que el particionamiento de mínimo diámetro es NP-duro para $M \geq 3$ o adaptarse para probar otros resultados de dureza NP. Considérese el grafo G_t al que se le ha añadido una arista. Si los vértices de esta arista no tienen el mismo color, o si la recoloración local (por ejemplo, por intercambio bicromático) da una coloración con no más colores que los previos, se puede proceder con el siguiente grafo. Cuando hay algún tipo de estructura en el conjunto O de estudio, esta se verá reflejada en los grafos G_t, que son más fáciles de colorear que los aleatorios. Se pueden resolver casos con $N \leq 600$.

Las particiones de mínimo diámetro no son únicas. De forma alternativa, se puede adaptar el algoritmo de coloración para encontrar una partición minimizando el segundo mayor diámetro de cluster sujeto a que el primero es mínimo,

después proceder al tercero de forma análoga, y así sucesivamente. Las particiones obtenidas mediante el algoritmo de enlace simple pueden sufrir en efecto de encadenamiento: entidades no similares pueden ser asignadas a un mismo cluster. Las particiones obtenidas mediante el algoritmo de coloración para mínimo diámetro pueden sufrir el llamado efecto de disección: entidades que sean similares pueden ser asignadas a diferentes clusters. Para evitar ambos efectos se pueden buscar soluciones que sean particiones eficientes para el criterio del split y del diámetro. El resultante algortimo con bicriterio para el Análisis Cluster se basa en las proposiciones 2.1 y 2.6. Imponer un valor mínimo en el split es suficiente para unir los vértices de G en extremos de aristas consecutivas del MST. El resultante grafo reducido G_R de G puede ser coloreado como se ha descrito anteriormente. Los splits y los diámetros de las particiones eficientes pueden ser representados gráficamente de forma que la representación pueda ser usada para evaluar si el conjunto O posee alguna estructura o no y qué particiones aparentan ser las más naturales. Una única partición eficiente por cada valor de M es una buena indicación.

En conclusión: Algunos algoritmos de clustering se aplican a grafos, que pueden ser vistos como grafos parciales G_t para un valor t (como el definido en el caso de la coloración, por ejemplo). Entonces, los clusters pueden ser definidos como componentes maximales con un grado mínimo de al menos δ, de forma que un algoritmo O($N + |E|$) proporciona una jerarquía de empaquetamientos que corresponden a los sucesivos valores de δ. Al igual que en el caso del clustering jerárquico, cuando se está aplicando clusteting a puntos de $\mathbb{R}^2$, hay propiedades geométricas que pueden ser explotadas para conseguir algoritmos polinomiales de bajo orden. Por ejemplo: el biparticionamiento de mínimo diámetro medio en el plano puede ser realizado en un tiempo O($Nlog^2N/loglogN$), y minimizar cualquier función monótona de los diámetros de una partición de M clusters puede ser realizado en un tiempo de O(N^{5M}).

2.3.3. Branch-and-Bound

Los algoritmos de branch-and-bound han sido aplicados con cierto éxito a varios problemas de particionamiento en el Análisis Cluster. Su eficiencia depende de cuánto se afine en las restricciones utilizadas, de la disponibilidad de buenas soluciones heurísticas y de una ramificación eficiente (reglas que mejoren las restricciones para todo subproblema obtenido).

Hay algoritmos de particionamiento para la mínima suma de cuadrados que

explotan cotas basadas en asignaciones de entidades a clusters ya construidos y la aditividad de las mismas para separar subconjuntos de entidades. Resuelven problemas con $N \leq 120$ y casos de clusters bien separados de puntos de $\mathbb{R}^2$, pero su rendimiento se deteriora según aunmenta la dimensión. Otros algoritmos, para el particionamiento de la mínima suma de cliques usan restricciones basadas en ordenar disimilitudes, aunque no son muy agudos. Se consiguen resolver problemas con $N \leq 50, M \leq 5$.

Mejores resultados se obtienen cuando las cotas se obtienen mediante la utilización de programación matemática. Para el particionamiento de la mínima suma de estrellas (problema de la M mediana) el algoritmo DUALOC combinado con relajación Lagrangiana en la cardinalidad de las restricciones es muy eficiente. Los problemas con $N \leq 900$ consiguen resolverse de forma exacta y la dimensión del espacio considerado no aparenta ser un problema en la resolución.

Surge una variante del problema de particionamiento de mínima suma de cliques cuando se busca una partición consenso (una partición que está a mínima distancia del conjunto de particiones dado). La distancia entre particiones viene medida mediante el número de pares de entidades en un mismo cluster en una partición y en diferentes clusters en la otra. Entonces, las disimilitudes pueden ser positivas o negativas, y el número de clusters no está fijado a priori. El problema puede ser expresado como sigue:

$$
\begin{array}{lll}
Mín & \displaystyle\sum_{k=1}^{N-1}\sum_{l=k+1}^{N} d_{kl}y_{kl} & \\
s.a: & y_{kl} + y_{lq} - y_{kq} \leq 1 & k = 1, 2, ..., N-2 \\
 & -y_{kl} + y_{lq} + y_{kq} \leq 1 & l = k+1, k+2, ..., N-1 \\
 & y_{kl} - y_{lq} + y_{kq} \leq 1 & q = l+1, l+2, ..., N \\
 & y_{kl} \in \{0, 1\} & k = 1, 2, ..., N-1 \qquad l = k+1, k+2, ..., N
\end{array}
$$

con:

$$
y_{kl} = \begin{cases} 1 & \text{si } O_k \text{ y } O_l \text{ pertenecen al mismo cluster} \\ 0 & \text{c.c.} \end{cases}
$$

Se consiguen resolver problemas con $N \leq 72$ aplicando el método símplex modificado al dual de la relajación continua de la formulación anterior. Una primera restricción a la suma de disimilitudes negativas es mejorada utilizando relaciones lógicas entre las variables y_{kl} (utilizando consecuencias de la desigualdad triangular). Por ejemplo: si $y_{kl} = 1$ entonces para todos los índices q, y_{kl} y y_{lq} son

iguales a 1 o ambos son iguales a 0 en cualquier solución factible. Si las variables son libres, la restricción se puede incrementar a:

$$\text{mín}\left\{ \text{máx}\{d_{kl}, 0\} + \text{máx}\{d_{lq}, 0\}, \text{máx}\{-d_{kq}, 0\} + \text{máx}\{-d_{lq}, 0\}\right\}$$

Se tienen en cuenta muchas consecuencias adicionales, consiguiendo restricciones bastante finas. Los casos con $N \leq 158$ pueden ser resueltos de forma más rápida que mediante un enfoque basado en los planos de corte, pero menos que mediante una combinación de técnicas heurísticas, planos de corte y branch-and-bound.

2.3.4. Métodos heurísticos

Para muchos criterios, alcanzar la solución exacta en un problema de clustering amplio está fuera de alcance, de modo que se recurre a la heurística. Encontrar una buena solución inicial puede ser importante en generación de columnas.

Las técnicas más tradicionales en heurística utilizan intercambio de entidades entre clusters o la redificinión de los clusters desde sus centroides. El algoritmo H-means, para el particionamiento de mínima suma de cuadrados, da una partición inicial aleatoria a partir de la cual procede al mejor intercambio de entidades de un cluster a otro hasta que se obtiene un mínimo local. El algoritmo K-means para el mismo problema también da lugar a una partición inicial aleatoria, pero a continuación computa los centroides de los clusters, asigna entidades a su centroide más cercano e itera hasta obtener un mínimo local. Ambos procedimientos se repiten un número de veces dado desde el comienzo. Ambos funcionan bien y dan buenos resultados cuando hay pocos clusters, pero se deterioran cuando hay muchos. Los experimentos muestran que el mejor clustering encontrado mediante K-means puede ser más de un $50\,\%$ peor que el mejor entre los conocidos.

Se obtienen mejores resultados aplicando metaheurísticos (simulated annealing, búsqueda Tabú, búsqueda genética,...). La reciente Búsqueda de Vecindad Variable (Variable Neighborhood Search) procede mediante búsqueda local a un mínimo local, después explora incrementando la distancia de las vecindades de esa partición mediante una perturbación al azar y realizando de nuevo búsqueda local. Se mueve a una nueva partición e itera si y sólo si se encuentra una mejor

que la actual. Los experimentos muestran que es un procedimiento muy eficiente para aproximar soluciónes de problemas de clustering con tamaño alto.

2.4. Otros procedimientos para el clustering

2.4.1. Clustering secuencial

Muchos algoritmos de clustering dan resultados sin tener en cuenta si el conjunto de entidades dado posee alguna estructura o no. No tiene en cuenta la posibilidad de ruido (entidades que sólo pueden ser clasificadas arbitrariamente). En esos casos puede ser preferible considerar los problemas de packing en lugar de los problemas de particionamiento. Se podría querer estudiar clusters de uno en uno, empezando por el más obvio, eliminando sus entidades e iterando. El clustering secuencial está cerca de los métodos de procesamiento de imagen:

Proceso: Encontrar clusters $C_k \subset O$ con $|C_k| = k = 1, 2, ..., |O|$ que optimicen un criterio. Evaluar el mejor valor k^* de k y la significación del cluster C_{k^*}. Si es significativo (mezcla entidades verdaderamente relacionadas) entonces O=$O \setminus \{C_{k^*}\}$ e iterar. Sino, parar.

En cada paso se resuelve un problema de clustering paramétrico en un sólo cluster, seguido de un test basado en la distribución de los valores del criterio. Algunos casos son fáciles: encontrar un cluster con máximo split puede ser realizado en un tiempo $\theta(N^2)$. Encontrar un cluster con mínimo radio o mínima estrella tiene un tiempo de O($N^2 logN$) si se ordenan las disimilutudes. Encontrar un cluster con mínimo diámetro es NP-duro, como también lo es encontrar un cluster con mínimo clique. El primer problema puede ser resuelto reduciéndolo a una secuencia de problemas de máximo clique, y el segundo expresándolo como un problema cuadrático de la mochila. También pueden considerarse criterios geométricos.

2.4.2. Clustering aditivo

Se pueden utilizar los clusters encontrados para explicar las disimilitudes (o similitudes) entre pares de entidades. Dada una matriz $S = (s_{kl})$ de similitudes entre pares de entidades de O, se buscan M clusters superpuestos $C_1, ..., C_M$ y sus correspondientes pesos $\lambda_1, ..., \lambda_M$ para minimizar la suma de cuadrados de los errores:

$$\sum_{k=1}^{N-1} \sum_{l=k+1}^{N} \left(s_{kl} - \sum_{\{j/O_k, O_l \in C_j\}} \lambda_j \right)^2$$

En una variante de este modelo, un cluster contiene a todas las entidades. Muchos heurísticos han sido propuestos para su solución, usando varias técnicas de programación matemática. Si se considera un cluster cada vez, con técnicas de análisis factorial el problema es mucho más sencillo y puede ser reducido a programación cuadrática o hiperbólica 0-1 con una restricción en cardinalidad.

Otro procedimiento a considerar podría ser la representación de disimilitudes mediante árboles, que pasa por considerar los dendrogramas obtenidos mediante algoritmos de agrupación jerárquica como árboles y ponderar las aristas mediante la disimilitud de las entidades que unen.

3. Un marco genérico para la selección de variables en los problemas de Análisis Clúster.

La solución del proceso de clustering dependía mucho de dos pasos en concreto: de la distancia a definir para el proceso y del criterio a seguir durante el mismo. Sin embargo, previo a esas dos elecciones, se forma la matriz de datos B a partir de las variables escogidas. Mediante Análisis Cluster se plantea el problema de clasificar a los individuos, pero siempre con respecto a la información disponible (información que es dada por las variables). Como ya se vió en la sección anterior, la presencia de variables relacionadas o que no aporten información relevante puede afectar de forma negativa al proceso de clustering (sobre todo, en aquellos problemas de alta dimensión), siendo necesario detectar las mismas para eliminarlas en el proceso de agrupamiento. Cuando no son detectadas y eliminadas, el análisis se ve influenciado por su presencia. La importancia de este problema aumenta según aumenta el número de variables. Hoy en día, las bases de datos pueden almacenar cientos e incluso miles de variables, siendo necesarias herramientas para seleccionar las más importantes (aquellas que mejor separen los clusters y descarten aquellas llamadas variables ruido).

En este apartado, se presenta una extensión del problema de la p-mediana, en el que la distancia entre entidades es calculada como la suma de las distancias en las q variables más importantes dentro de un conjunto de tamaño m. Este modelo tiene la aplicación buscada para el Análisis Cluster, puesto que de una amplia lista de variables originales para el proceso, sólo un subconjunto de ellas es apropiado para encontrar la estructura grupal de los datos. En la práctica se partirá de un conjunto inicial de m variables de las cuales se seleccionarán las p que mejor explican dicha estructura, dando lugar a las p variables consideradas para la sección anterior en el proceso de Análisis Cluster. Así, los investigadores pueden separar dichas variables de las otras antes de realizar las particiones en los datos.

Este problema puede ser formulado como un problema de optimización no lineal entera mixta donde el agrupamiento y la selección de variables se realizan de forma simultánea. A continuación, se proponen dos linealizaciones distintas y se comparan sus rendimientos con el método de clustering por defecto con todas las variables (problema de la p-mediana), mostrando que el modelo basado en formulación radial resulta el mejor.

3.1. Introducción a la selección de variables

Se pretende estudiar el siguiente problema de agrupamiento: Supongamos dado un conjunto $O = \{O_1, ..., O_N\}$ de unidades estadísticas (entidades) medidas respecto a un conjunto de características cualitativas o cuantitativas
$V = \{v_1, ..., v_m\}$. Esta información se representa en una matriz de datos
$B = (O_{ik})$, donde O_{ik} representa el valor de la característica v_k en la entidad O_i. El objetivo es encontrar un subconjunto de variables $Q \subseteq V$ de tamaño q prefijado y agrupar las N entidades en M clusters de forma que el clustering resultante de O es el más preciso cuando sólo se cuenta con la información de las variables de Q para decidir los clusters.

Este problema surge en Estadística, donde ya es conocido el hecho de que no todas las variables son igualmente importantes a la hora de encontrar la estructura grupal de los datos. Esto es debido a que incluir todas las variables deteriora la efectividad del procedimiento de agrupación, llegando incluso a obtener una clasificación errónea. Aquellas variables que no definen la estructura en grupos de los datos son denominadas variables ruido.

En el presente trabajo se propone un modelo combinatorio para el agrupamiento que selecciona simultáneamente el mejor subconjunto $Q \subseteq V$ de variables, el mejor conjunto de medianas $P \subseteq O$, y la partición óptima de datos cuando el criterio utilizado es minimizar la distancia total dentro de los clusters entre la mediana del cluster y las entidades que pertenecen al mismo. Puede verse como una extensión del problema de la p-mediana (haciendo $p = M$), donde las distancias dependen de las variables Q seleccionadas. Se formula como sigue:

$$\min_{\{P \subseteq O, Q \subseteq V / |P|=p, |Q|=q\}} \sum_{O_i \in O} \min\{d_{ij}^Q / O_j \in P\}$$

donde d_{ij}^Q es la distancia entre las unidades O_i, O_j restringida a las variables en Q.

Para esta sección, se trabajará con la distancia Manhattan (o distancia l_1), una de las mas comunes a la hora de agrupar datos ordinales o cualitativos. El modelo aquí propuesto es un método exacto que cambia de una función objetivo procedente del algoritmo k-means a una función objetivo relativa a la p-mediana. El enfoque basado en la p-mediana tiene varias ventajas en términos de robustez e interpretación, tales como que la mediana representante del cluster es un elemento de la muestra.

El modelo de la p-mediana puede extenderse para considerar la decisión de

qué variables $Q \subseteq V$ seleccionar, pero la formulación natural de esta extensión lleva a un problema cuadrático no convexo. En lugar de desarrollar herramientas para solucionar este modelo no lineal, nos centramos en estudiar distintas linealizaciones enteras mixtas y determinar cuál es la más eficiente. La primera será una linealización directa del modelo cuadrático mientras que la segunda se basa en la llamada formulación radial del problema de la p-mediana, ambas propuestas por los autores Stefano Benati y Sergio García.

3.2. Definición del problema y formulación

Supongamos dada una muestra $O = \{O_1, ..., O_N\}$ de unidades estadísticas. Para cada entidad O_i, se mide el conjunto de variables $V = \{v_1, ..., v_m\}$. Supongamos que las variables v_k están representadas por datos ordinales o cualitativos. En el caso de que los datos sean cualitativos, se representan mediante $0 - 1$. En el caso de que sean ordinales con g posibles valores o de que estén representados mediante una escala de Likert con un número g de niveles, entonces g se denominará dimensión de la escala.

Sea O_{ik} el valor que toma la variable v_k en la entidad O_i. La distancia (o diferencia) entre las entidades O_i y O_j con respecto a la característica v_k es $d_{ijk} = |O_{ik} - O_{jk}|$, y la distancia total entre ambas entidades viene dada por la norma 1:

$$d_{ij} = \sum_{k=1}^{m} d_{ijk} = \sum_{k=1}^{m} |O_{ik} - O_{jk}|$$

Supongamos ahora que sólo un subconjunto $Q \subseteq V$ de variables son consideradas relevantes para el análisis y que, en consecuencia, las diferencias entre las entidades son calculadas utilizando sólamente el conjunto Q. La distancia se expresa mediante el vector de incidencia z sobre el conjunto Q:

$$d_{ij} = \sum_{k=1}^{m} d_{ijk} z_k$$

donde $z_k = 1$ si $v_k \in Q$ y $z_k = 0$ en caso contrario.

Las entidades son agrupadas usando el modelo de la p-mediana y el criterio de mínima suma, así que el resultado serán M clusters y su mediana será el elemento más representativo.

Se definen variables binarias y_j con $j = 1, ..., N$ de la siguiente forma:

$$y_j = \begin{cases} 1 & \text{si } O_j \text{ es una mediana} \\ 0 & \text{c.c.} \end{cases}$$

y también variables binarias de asignación x_{ij}, con $i, j = 1, ..., N$:

$$x_{ij} = \begin{cases} 1 & \text{si } O_i \text{ es asignada al cluster } C_j \\ 0 & \text{c.c.} \end{cases}$$

Para imponer que sólo se utilice un subconjunto $Q \subseteq V$, de variables, se definen las variables z_k, con $k = 1, ..., m$:

$$z_k = \begin{cases} 1 & \text{si } v_k \in Q \\ 0 & \text{c.c.} \end{cases}$$

Con estas variables, el modelo obtenido es el siguiente:

$$(F_1) \begin{cases} Mín. & \displaystyle\sum_{i=1}^{N}\sum_{j=1}^{N}\Big(\sum_{k=1}^{m} d_{ijk}z_k\Big)x_{ij} & \\ s.a: & x_{ij} \leq y_j & i, j = 1, ..., N \\ & \displaystyle\sum_{j=1}^{N} x_{ij} = 1 & i = 1, ..., N \\ & \displaystyle\sum_{j=1}^{N} y_j = M & \\ & \displaystyle\sum_{k=1}^{m} z_k = q & \\ & x_{ij} \geq 0 & i, j = 1, ..., N \\ & y_j \in \{0, 1\} & j = 1, ..., N \\ & z_k \in \{0, 1\} & k = 1, ..., m \end{cases}$$

Esta formulación es no lineal debido a los términos cuadráticos en la función objetivo. De hecho, la función objetivo no es convexa debido a que la matriz de distancias no es semidefinida positiva (los términos d_{ijk} pueden disponerse de forma que la matriz esté compuesta por 0 en la diagonal principal y el resto de términos extradiagonales sean 0 ó positivos). Para un estudio más sencillo del problema, se procede a una linealización de la formulación F_1.

Puede observase que, cuando todas las variables z_k toman el valor 1, se obtiene el problema de la p-mediana.

3.2.1. Linealización directa

La forma más sencilla de proceder a la linealización es introducir nuevas variables:

$w_{ijk} = x_{ij} z_k, \quad i, j = 1, ..., N; \; k = 1, ..., m$

además de nuevas restricciones de buena definición. El modelo obtenido es el siguiente:

$$
(F_2) \begin{cases}
M\acute{\imath}n. & \displaystyle\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{m} d_{ijk} w_{ijk} & & & & \\
s.a: & x_{ij} \leq y_j & i, j = 1, ..., N & & & \\
& \displaystyle\sum_{j=1}^{N} x_{ij} = 1 & i = 1, ..., N & & & \\
& \displaystyle\sum_{j=1}^{N} y_j = M & & & & \\
& \displaystyle\sum_{k=1}^{m} z_k = q & & & & \\
& w_{ijk} \geq x_{ij} + z_k - 1 & i, j = 1, ..., N & k = 1, ..., m & & \\
& w_{ijk} \leq x_{ij} & i, j = 1, ..., N & k = 1, ..., m & (a) & \\
& w_{ijk} \leq z_k & i, j = 1, .., N & k = 1, ..., m & (b) & \\
& w_{ijk} \geq 0 & i, j = 1, ..., N & k = 1, ..., m & & \\
& x_{ij} \geq 0 & i, j = 1, ..., N & & & \\
& y_j \in \{0, 1\} & j = 1, ..., N & & & \\
& z_k \in \{0, 1\} & k = 1, ..., m & & &
\end{cases}
$$

No se está imponiendo que las variables x_{ij} deban ser binarias, sólo positivas. Esto es debido a que se tiene un problema de minimización en el que las distancias d_{ijk} son positivas, además de que y_j y z_k son binarias. En consecuencia, existe una solución óptima donde todas las variables x_{ij} son binarias. Además, las inecuaciones (a) y (b) podrían eliminarse porque se satisfacen en cada solución óptima.

3.2.2. Formulación radial

A continuación se propone otro modelo también aplicado al problema de la p-mediana con el que se obtienen resultados satisfactorios. Dada una entidad O_i y una variable v_k, se tiene que como las variables están expresadas mediante la escala Likert con g niveles, muchas entidades están localizadas a una misma distancia (es decir, a un mismo radio). Para obtener la formulación radial, se procede como sigue:

- **Paso 1:** Dado un cliente i, ordenar las distancias $\{d_{i1k}, ..., d_{iNk}\}$ en orden creciente y eliminar multiplicidades para obtener G_{ik} valores distintos. Si D_{ik1} es el menor coste, D_{ik2} es el segundo menor coste, y así, se tiene:

  $0 = D_{ik1} < D_{ik2} < ... < D_{ikG_{ik}}$

  Además, también se tiene que $G_{ik} \leq g$.

- **Paso 2:** Definir variables binarias r_{ikt} de la siguiente forma:

$$r_{ikt} = \begin{cases} 1 & \text{si seleccionada } v_k, O_i \text{ se encuentra al menos a distancia } D_{ikt} \\ 0 & \text{c.c} \end{cases}$$

Se puede ver que:

$$\sum_{t=2}^{G_{ik}} (D_{ikt} - D_{ik(t-1)}) r_{ikt} = \sum_{j=1}^{N} d_{ijk} w_{ijk}$$

y en consecuencia, la función objetivo se puede expresar como:

$$\sum_{i=1}^{N} \sum_{k=1}^{m} \sum_{t=2}^{G_{ik}} (D_{ikt} - D_{ik(t-1)}) r_{ikt}$$

El modelo resultante utiliza también variables y_j, x_{ij} y z_k definidas como en los modelos anteriores:

$$
(F_3)\begin{cases}
\textit{Mín.} & \displaystyle\sum_{i=1}^{N}\sum_{k=1}^{m}\sum_{t=2}^{G_{ik}}(D_{ikt}-D_{ik(t-1)})r_{ikt} & & \\
s.a: & x_{ij}\leq y_j & i,j=1,...,N & (c)\\
& \displaystyle\sum_{j=1}^{N} x_{ij}=1 & i=1,...,N & \\
& \displaystyle\sum_{j=1}^{N} y_j = M & & \\
& \displaystyle\sum_{k=1}^{m} z_k = q & & \\
& r_{ikt}+\displaystyle\sum_{\{j/d_{ijk}<D_{ikt}\}} x_{ij}\geq z_k & i=1,...,N \quad k=1,...,m & \\
& & t=2,...,G_{ik} & (d)\\
& r_{ikt}\geq 0 & i=1,...,N \quad k=1,...,m & \\
& & t=2,...,G_{ik} & \\
& x_{ij}\geq 0 & i,j=1,...,N & \\
& y_j\in\{0,1\} & j=1,...,N & \\
& z_k\in\{0,1\} & k=1,...,m &
\end{cases}
$$

Dados una entidad O_i, una característica v_k y un nivel de distancia D_{ikt}, la restricción (d) impone lo siguiente: Si la característica v_k es seleccionada ($z_k = 1$), entonces o bien O_i se asigna a una mediana O_j que se encuentra a una distancia $d_{ijk} < D_{ijt}$, ó, si no es posible, se asigna a una distancia al menos D_{ikt} ($r_{ikt} = 1$).

Al igual que en el modelo anterior, como se está minimizando y las distancias son positivas, existe una solución óptima donde todas las variables x_{ij} son binarias. De igual forma, puede verse que no es necesario imponer que las variables r_{ikt} sean binarias debido a que tendrán valor 0 ó 1 en cualquier solución óptima.

Comparando ambas linealizaciones (F_2 y F_3), se observa que ambas tienen $N+m$ variables binarias. Sin embargo, F_2 tiene N^2m+N^2+N+2 restricciones y N^2m+N^2+N+m variables, mientras que F_3 tiene como mucho $Nmg+N^2+N+2$ restricciones y $Nmg+N^2+N+m$ variables. Como g es mucho menor que N, F_3 tiene un menor número de restricciones y variables que F_2.

3.3. Problema adicional con centros prefijados

A continuación, se expone un caso particular del problema de selección de variables para el clustering. Análogo al problema anterior, se formula como un problema de optimización con variables binarias que representan la decisión de seleccionar o rechazar: Se dan un conjunto O de unidades estadísticas (entidades) y un conjunto R de centros de clusters prefijados que pueden haberse obtenido mediante un procedimiento previo (como una etapa previa del algoritmo K-means) o por propia elección del investigador. Ambos son medidos a través de otro conjunto de variables V, y una función de disimilitud $d(i, j, k)$ para cada $i \in O, j \in R, k \in V$. El objetivo consiste en encontrar un subconjunto de variables $Q \subseteq V$ tal que el total de las distancias de las unidades al centro más cercano sea mínima.

El problema puede ser interpretado de la siguiente forma: conociendo los centros de cluster prefijados (y por tanto, el número de clusters que se generarán durante el proceso de clustering) se quiere conocer el subconjunto de variables que da lugar a esa agrupación, ya sea porque se ha obtenido como mejor solución en un procedimiento previo de clustering (mediante métodos aglomerativos, divisivos, o heurísticos) o porque se fijan los centros en función del interés del investigador.

Pese a que este problema es NP-completo, se proponen algoritmos eficientes y exactos, además de heurísticos. El método heurístico propuesto opera de forma similar al clásico algoritmo K-means, obteniendo la solución óptima en pocas iteraciones. Para los algoritmos exactos se experimenta con distintas formulaciones en programación lineal mixta, llegando a que la mejor es la formulación radial (como se obtuvo en el caso anterior). Su relajación continua es cerana al óptimo y la solución entera se calcula en pocos segundos. Cuando estos algoritmos se aplican al clustering, la calidad de las soluciones obtenidas mejora considerablemente.

3.3.1. Formulación del problema

El siguiente modelo determina las asignaciones óptimas y discriminación de variables para este problema de clustering:

Dados un conjunto $O = \{O_1, ..., O_N\}$ de unidades estadísticas (entidades) y un conjunto $R = \{R_1, ..., R_r\}$ de centros de clusters o prototipos, para cada $i \in O$ y cada $j \in R$ se mide un conjunto $V = \{v_1, ..., v_m\}$ de variables; de forma que para cada triplete $i \in O$, $j \in R$ y $k \in V$ se mide una distancia d_{ijk} representando la disimilitud entre la entidad O_i y el centro R_j de acuerdo a la variable v_k. Si un

subconjunto $Q \subseteq V$ de variables es seleccionado, entonces la distancia entre O_i y R_j usando Q tiene la siguiente expresión:

$$d_{ij}(Q) = \sum_{v_k \in Q} d_{ijk}$$

Las distancias que usan este conjunto Q se usan para determinar una partición de O en clusters $C_j,\ j = 1, ..., r$. Para un $Q \subseteq V$ fijado, una entidad O_i es asignada al cluster $C_{j(i)}$ si

$$d_{i,j(i)}(Q) = \text{mín}\{d_{ij}(Q) | j = 1, ..., r\}$$

La distancia total entre unidades y clusters es la suma

$$D(Q) = \sum_i d_{i,j(i)}(Q)$$

Si este valor se lleva a un mínimo (en analogía a la metodología del algoritmo K-means) entonces $D(Q)$ es un índice de la calidad de la partición inducida por Q. El objetivo es seleccionar el subconjunto $Q \subseteq V$ con cardinalidad $|Q| = q$ para el cual el índice $D(Q)$ se minimiza. Este problema se llamará Selección de q Variables.

Para el modelo, se requiere que el número q de variables seleccionadas sea fijado previamente, aunque los investigadores no tienen en cuenta este parámetro. Como ocurrre para el algoritmo K-means, este modelo se ejecuta para distintos valores de q y se selecciona aquel que produzca un mayor cambio en la función objetivo cuando se pasa de q a $q + 1$. Si el modelo se aplica dentro del modelo de clustering, entonces q se puede seleccionar a traves del llamado BIC: Criterio de Información Bayesiano.

El índice $D(Q)$ está estrechamente relacionado con minimizar la variabilidad dentro de los clusters. Si las variables están estandarizadas, las disimilitudes d_{ijk} son distancias al cuadrado, y los centroides R se calculan como la media de los clusters; entonces la función objetivo $D(Q)$ corresponde a minimizar la variabilidad dentro de los grupos, lo cual es equivalente a maximizar la variabilidad entre distintos grupos. Veamos esto con más detalle:

Como ya se introdujo en la sección anterior, sea O_{ik} el valor de la variable v_k medida en la entidad O_i, y sea $\mu_k = \frac{1}{N} \sum_{i=1}^{N} O_{ik}$ la media de la variable v_k.

La variabilidad total de v_k se expresa mediante la siguiente suma de cuadrados:

$$TSS_k = \sum_{i=1}^{N}(O_{ik} - \mu_k)^2$$

Considérese el caso en que las entidades están particionadas en clusters C_j con $j = 1, ..., r$, y cada centro de cluster está representado por su media. Es decir, la k-ésima coordenada de C_j es $r_{jk} = \frac{1}{|C_j|} \sum_{O_i \in C_j} O_{ik}$. Sea $C_{j(i)}$ el cluster al que la entidad O_i ha sido asignada. Para la partición dada, la suma total de cuadrados es la siguiente:

$$TSS_k = \sum_{i=1}^{N}(O_{ik}-\mu_k)^2 = \sum_{i=1}^{N}(O_{ik}-r_{j(i),k})^2+\sum_{i=1}^{N}(r_{j(i),k}-\mu_k)^2+2\sum_{i=1}^{N}(O_{ik}-r_{j(i),k})(r_{j(i),k}-\mu_k)$$

El término $\sum_{i=1}^{N}(O_{ik} - r_{j(i),k})(r_{j(i),k} - \mu_k)$ es nulo. De forma más detallada:

En primer lugar:

$$\sum_{i=1}^{N} O_{ik}\mu_k = N(\mu_k)^2$$

Por otro lado:

$$\sum_{i=1}^{N} \mu_k r_{j(i),k} = \mu_k \sum_{j=1}^{r} |C_j| r_{jk} = \mu_k \sum_{j=1}^{r} \Big(\sum_{O_i \in C_j} O_{ik} \Big) = \mu_k \sum_{i=1}^{N} O_{ik} = N(\mu_k)^2$$

Estos dos se cancelan al desasollar el producto. En los términos restantes se obtiene:

$$\sum_{i=1}^{N} O_{ik} r_{j(i),k} = \sum_{j=1}^{r} r_{jk} \Big(\sum_{O_i \in C_j} O_{ik} \Big) = \sum_{j=1}^{r} r_{jk}|C_j| r_{jk} = \sum_{j=1}^{r} |C_j|(r_{jk})^2$$

que cancela con el siguiente:

$$\sum_{i=1}^{N} r_{j(i),k} r_{j(i),k} = \sum_{j=1}^{r} |C_j| (r_{jk})^2$$

al desarrollar en el producto.

En consecuencia, la suma total de cuadrados puede descomponerse en dos términos:

$$WSS_k = \sum_{i=1}^{N} (O_{ik} - r_{j(i),k})^2$$

representando la variabilidad dentro de los clusters

$$CSS_k = \sum_{i=1}^{N} (r_{j(i),k} - \mu_k)^2$$

representando la variabilidad entre los distintos clusters

Al escoger un conjunto $Q \subseteq V$ de variables la descomposición de variabilidad depende del conjunto Q de acuerdo con la siguiente expresión:

$$\sum_{v_k \in Q} TSS_k = \sum_{v_k \in Q} WSS_k + \sum_{v_k \in Q} CSS_k$$

Sin embargo, calcular $\min_{Q \subseteq V} \sum_{v_k \in Q} CSS_k$ fijando un conjunto Q de tamaño q no se corresponde con calcular $\max_{Q \subseteq V} \sum_{v_k \in Q} WSS_k$, puesto que $\sum_{v_k \in Q} TSS_k$ depende de Q. Sólo en el caso concreto en que las variables V estén estandarizadas ambos problemas son equivalentes.

Teorema 3.1. *Supongamos que las variables V se miden en unidades o entidades O y que todas las variables tienen la misma varianza $\sigma_k^2 = \sigma^2$ para todo $v_k \in V$. Sean C_j, $j = 1, ..., r$ los clusters en los que las unidades son particionadas, y sea R el conjunto de centros de clusters calculados como la media de los clusters, de la misma forma que se describió anteriormente. Con estos datos, resuélvase la Selección de q Variables con $d_{ijk} = (O_{ik} - r_{j(i),k})^2$. Si se alcanza una solución*

óptima Q^* *tal que* $\sum_{v_k \in Q^*} CSS_k = \min_{\{Q \subseteq V / |Q| = q\}} \sum_{v_k \in Q} CSS_k$*, entonces* $\sum_{v_k \in Q^*} WSS_k = \max_{\{Q \subseteq V / |Q| = q\}} \sum_{v_k \in Q} WSS_k$.

Demostración: Bajo la condición $\sigma_k^2 = \sigma^2$ para todo $v_k \in V$, se tiene que:

$$TSS_k = \sum_{i=1}^{N} (O_{ik} - \mu_k)^2 = N\sigma^2$$

lo cual implica:

$$\sum_{v_k \in Q} TSS_k = qN\sigma^2$$

Es decir, TSS_k es constante. Teniendo en cuenta la descomposición $\sum_{v_k \in Q} TSS_k = \sum_{v_k \in Q} WSS_k + \sum_{v_k \in Q} CSS_k$ es inmediato observar que, bajo estas condiciones, según aumenta un sumando el otro debe disminuir. En consecuencia, se tiene el resultado. □

Es conocido el hecho de que el clustering con selección de variables es NP-completo, puesto que contiene al problema de la p-mediana. Pero, incluso en el caso en que los centros de clusters están fijados, el problema resultante (Selección de q Variables) no es resoluble en tiempo polinomial. Los autores Stefano Benati, Sergio García y Justo Puerto prueban en [2] este hecho, cuya prueba radica en el hecho de que la selección de q variables contiene el problema de la p-mediana:

El problema de la selección de q variables puede plantearse como sigue: Dada una matriz de distancias $D \in \mathbb{R}^{N \times r \times m}$ y un número real no negarivo α, ¿existe una selección de variables Q tal que la signación resultante de clusters tenga un valor $g^* \leq \alpha$?

Por otro lado, considérese un problema p-mediana con distancias c_{ij} para clientes $i \in A$, con $|A| = N$ y localizaciones $j \in B$ con $|B| = m$. Para un conjunto de medianas $P \subseteq B$ con $|P| = p$, la distancia de $i \in A$ a P es $c_{i,P} = \min\{c_{ij} | j \in P\}$. Entonces, el problema de la p-mediana se formula como $\min_{P \subseteq B} \sum_{i \in V} d_{i,P}$. Se sabe que este problema, que puede plantearse como sigue a continuación, es NP-completo:

Problema de la p-mediana: Dada una matriz de coste $C \in \mathbb{R}^{N\times m}$ y un valor real no negativo α, ¿existe una p-mediana de valor $v^* \leq \alpha$?

Teorema 3.2. *La Selección de q Variables es un problema NP-completo*

Demostración: Comprobar si una solución Q es tal que la función objetivo verifica $g^* \leq \alpha$ puede ser realizado en tiempo polinomial, por tanto el problema se encuentra en NP. Para ver la completitud, mostraremos que el problema de la p-mediana puede ser reducido a la selección de q variables, con $p = q$.

Dado un problema de la p-mediana con matriz de costes $C \in \mathbb{R}^{N\times m}$ el problema q-variable selection con matriz de costes $D \in \mathbb{R}^{N\times m\times m}$ se define en el grafo auxiliar bipartito $G = (A, B, E)$, en el que los elementos $i \in A$ representan a los clientes y B es tal que $|B| = m$. Para cada $i \in A$, $j \in \{1, ..., m\}$ hay m arcos indexados por $k \in K = \{1, ..., m\}$, cuyos pesos son:

$$d_{ijk} = \begin{cases} c_{ik} & \text{si } j = k \\ M' & \text{c.c.} \end{cases}$$

donde M' es un escalar suficientemente grande. La estructura de la matriz tridimensional D es la siguiente (con d^j representando la matriz de distancias para $i \in A,\ j \in B$):

$$\begin{array}{c|cccc|cccc|c|cccc|} & \overbrace{\quad}^{d^1} & & & & \overbrace{\quad}^{d^2} & & & & & \overbrace{\quad}^{d^m} & & & \\ 1 & c_{11} & M' & \dots & M' & M' & c_{12} & \dots & M' & \dots & M' & \dots & M' & c_{1m} \\ 2 & c_{21} & M' & \dots & M' & M' & c_{22} & \dots & M' & \dots & M' & \dots & M' & c_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ N & c_{N1} & M' & \dots & M' & M' & c_{N2} & \dots & M' & \dots & M' & \dots & M' & c_{Nm} \end{array}$$

Puede observarse que en este problema de selección de variables hay m variables, cada una de ellas medida con respecto a m centros de cluster. Ahora, considérese la selección de q variables en este grafo, con $q = p$. Siempre que un conjunto $P \subseteq B$ con $|P| = p$ se selecciona, la distancia entre la unidad $i \in A$ y la característica $j \in \{1, ..., m\}$ de acuerdo a las variables seleccionadas en P es

$$d_{ij} = \begin{cases} d_{ikk} + (p-1)M' = c_{ik} + (p-1)M' & \text{si } j = k \in P \\ pM' & \text{c.c.} \end{cases}$$

En consecuencia, la asignación de decisiones es:

$$x_{ij} = \begin{cases} 1 & \text{si } d_{ij} = \text{mín}\{d_{iw} | w \in P\} \\ 0 & \text{c.c.} \end{cases}$$

Pero para $x_{ij} = 1$, utilizando que $c_{iw} + (p-1)M' < pM'$, se tiene $d_{ij} = \text{mín}\{c_{iw} + (p-1)M' | w \in P\} = \text{mín}\{c_{iw} | w \in P\} + (p-1)M'$.

Entonces, la función objetivo del problema de selección de variables es $\sum_{i,j} d_{ij}x_{ij} = \sum_{i} \text{mín}\{c_{iw} | w \in P\} + N(p-1)M'$.

Por otro lado, la función objetivo del problema de la p-mediana es $\sum_{i} \text{mín}\{c_{iw} | w \in P\}$.

Se obtiene, por tanto, que la selección de q variables contiene al problema de la p-mediana. Entonces, con $q = p$, el problema de la p-mediana tiene solución con valor $v^* < \alpha$ si y sólo si la selección de q variables tiene, en el grafo auxiliar, una solución con valor $g^* < \alpha + N(p-1)M'$. Como el problema de la p-mediana es NP-completo, se tiene también la completitud para la selección de q variables. □

En consecuencia, se establece que no existe un algoritmo de tiempo polinomial que resuelva la selección de q variables a menos que P=NP.

Con objeto de desarrollar métodos exactos y heurísticos que calculen la solución óptima del problema, se formula el mismo como un problema en programación lineal entera mixta (MILP).

3.3.2. Formulación MILP

El conjunto $Q \subseteq V$ óptimo para el q-variable selection problem puede ser determinado a través de un problema en programación lineal entera mixta (MILP). Para dicho problema, se tienen como variables:

- Variables de selección de distancia z_k, definidas como:

$$z_k = \begin{cases} 1 & \text{si } v_k \in Q \\ 0 & \text{c.c.} \end{cases}$$

- Variables de asignación (global) x_{ij} de la unidad O_i al cluster C_j, mediante las que se tiene:

$$x_{ij} = \begin{cases} 1 & \text{si } O_i \text{ asignada al cluster } C_j \\ 0 & \text{c.c.} \end{cases}$$

- Variables de asignación (local) w_{ijk} de la unidad O_i al cluster C_j usando la diferencia a través de v_k, mediante las que se tiene:

$$w_{ijk} = \begin{cases} 1 & \text{si } O_i \text{ es asignada a } C_j \text{ y } v_k \text{ es seleccionada} \\ 0 & \text{c.c} \end{cases}$$

El modelo resultante MILP es el siguiente:

$$(P_1) \begin{cases} \text{Mín.} & \displaystyle\sum_{i=1}^{N}\sum_{j=1}^{r}\sum_{i=1}^{m} d_{ijk} w_{ijk} & & \\ \text{s.a:} & \displaystyle\sum_{k=1}^{m} w_{ijk} = q x_{ij} & \forall i,\ \forall j & (1) \\ & \displaystyle\sum_{j=1}^{r} x_{ij} = 1 & \forall i & (2) \\ & \displaystyle\sum_{j=1}^{r} w_{ijk} \leq z_k & \forall i,\ \forall k & (3) \\ & \displaystyle\sum_{k=1}^{m} z_k = q & & (4) \\ & w_{ijk} \in \{0,1\} & \forall i,\ \forall j,\ \forall k & (5) \\ & x_{ij} \in \{0,1\} & \forall i,\ \forall j & (6) \\ & z_k \in \{0,1\} & \forall k & (7) \end{cases}$$

Las restricciones establencen lo siguiente:

(1) establece que no es factible una asignación local de O_i a C_j usando la variable v_k a menos que se establezca una asignación global O_i a C_j. También indica que el número total de asignaciones locales de O_i a C_j es q.

(2) establece que cada entidad O_i debe ser asignada a un cluster C_j exactamente.

(3) impone que una asignación local de O_i a C_j usando la variable v_k es factible si y sólo si se selecciona la variable v_k.

Denotemos como z^{IP_1} al valor óptimo del problema P_1, y como z^* al valor óptimo de la relajación de P_1 al problema en que (5) y (7) son relajadas a continuidad de forma que $0 \leq w_{ijk} \leq 1$ para toda posible combinación de i, j, k y $0 \leq z_k \leq 1$ para todo k. Entonces:

Teorema 3.3. $z^{IP_1} = z^*$.

Demostración: Para una solución factible x, sea $R_{j(i)}$ el centro de cluster R_j para el cual $x_{i,j(i)} = 1$. Entonces, P_1 se simplifica de la siguiente forma:

$$(P_2) \begin{cases} Mín & \displaystyle\sum_{i=1}^{N} \left(\sum_{k=1}^{m} d_{i,j(i),k} w_{i,j(i),k} \right) & & \\ \text{s.a:} & \displaystyle\sum_{k=1}^{m} w_{i,j(i),k} = q & \forall i & (8) \\ & w_{i,j(i),k} \leq z_k & \forall i,\ \forall k & (9) \\ & \displaystyle\sum_{k=1}^{m} z_k = q & & (10) \\ & w_{i,j(i),k} \in \{0,1\} & \forall i,\ \forall k & (11) \\ & 0 \leq z_k \leq 1 & \forall k & (12) \end{cases}$$

Debido a las restricciones (8) y (10), cada restricción en (9) se satisface como igualdad: supongamos que existe $\hat{k}$ para la cual (11) es una desigualdad estricta. Si se suman todas las restricciones (9) sobre el índice k, se obtiene $q < q$, lo cual es una contradicción.

Usando esta propiedad, la función objetivo puede escribirse como:

$$\sum_{k=1}^{m} \Big(\sum_{i=1}^{N} d_{i,j(i),k} \Big) z_k$$

Esta función objetivo depende únicamente de la restricción de cardinalidad impuesta en (10) y de que z sea binaria. Llamemos $b_k = \sum_{i=1}^{N} d_{i,j(i),k}$ y ordenemos los valores b de forma que $b_{i(1)} \leq b_{i(2)} \leq \cdots \leq b_{i(m)}$. Entonces z puede ser relajada de forma que se encuentre entre 0 y 1, puesto que existe una solución de P_2 tal que $z_k = 1$ si y sólo si $k = i(j)$ para algún $j \leq q$. Entonces, para valores enteros de z y x existe una solución w con valores enteros. Con esto, se tiene el resultado. □

El problema P_2 puede interpretarse de la siguiente forma: Para un conjunto de entidades O, centros de clusters R y asignaciones O_i a $R_{j(i)}$ con $O_i \in O$, $R_{j(i)} \in R$, encontrar un conjunto $Q \subseteq V$ tal que la suma de las distancias de O_i a $R_{j(i)}$ sea minimizada. La construcción anterior muestra que el problema restringido es resoluble en tiempo polinomial.

3.3.3. Restricciones adicionales

La formulación del problema con un modelo MILP permite manejar restricciones adicionales que pueden ser impuestas por el investigador. Entre los objetivos de las mismas, pueden encontrarse:

- Restricción en variabilidad total.
- Restringir covarianzas.
- Equilibrio en la cardinalidad de los clusters.
- Descartar observaciones outlier.

Siempre y cuando estas restricciones sean lineales, pueden ser añadidas a P_1 sin aumentar su dificultad teórica. En consecuencia, es probable que los algoritmos propuestos para la resolución de P_1 puedan extenderse sin dificultad a los casos en que se añaden estas restricciones. No son necesarias para la formulación estándar del problema, sino una elección propia del investigador.

3.3.4. Formulación radial

El cálculo de la solución z^{IP_0} requiere un modelo MILP con un orden cuadrático de variables binarias, que son las asignaciones i, j. Esto puede ser innecesario, ya que el objetivo del problema y la decisión del mismo son las variables a seleccionar. En estas circunstancias, se explora la posibilidad de conseguir una formulación alternativa que pueda hacer disminuir el número de variables binarias utilizadas.

La formulación radial es una técnica que consigue escribir la función objetivo de un problema MILP como una suma telescópica de términos. Muchos de estos términos son redundantes a la hora de efectuar el cálculo de la función objetivo, por lo que la formulación radial supone una vía más rápida a la hora de resolver el problema original.

Esta formulación reemplaza las variables de asignación w_{ijk} por variables radiales r_{ijk}. Para definir estas nuevas variables, se procede de la siguiente forma: Partiendo de cualquier par (O_i, v_k), se realizan de forma análoga al problema expuesto en la sección 3.2.2 los siguientes pasos:

- **Paso 1:** Ordenar $\{d_{i1k}, d_{i2k}, ..., d_{iNk}\}$ y eliminar multiplicidades hasta obtener:

$$D_{i1k} < D_{i2k} < ... < D_{i,g(i,k),k}$$

- **Paso 2:** Definir variables binarias h_{itk} como sigue:

$$h_{itk} = \begin{cases} 1, & \text{si } O_i \text{ se asigna a un centro a distancia al menos } D_{itk} \\ 0, & \text{c.c.} \end{cases}$$

La formulación radial puede hacer decrecer el tamaño del problema: Dados $O_i \in O$, $v_k \in V$ y una constante $c \in \mathbb{R}$, se definen niveles como los conjuntos $T_c^{ik} = \{R_j \in R | d_{ijk} = c\}$. En el paso 1, si R_j, $R_z \in T_c^{ik}$ para algún c, entonces existe un índice u tal que $D_{iuk} = d_{ijk} = d_{izk}$, siendo $g(i,k)$ el número total de niveles no vacíos. En el peor de los casos, todas las distancias son distintas, y por ende r, (número de clusters) es igual a $g(i,k)$ (número de niveles en el paso 1). Sin embargo, cuando esto no ocurre, la formulación radial elimina las multiplicidades de los datos (distancias repetidas) y el número de variables decrece. Esto puede tener un impacto importante en el caso de medir variables cualitativas 0,1.

Para cada par dado $O_i \in O$, $v_k \in V$, el correspondiente término de la función objetivo puede ser reescrito de la forma:

$$\sum_{t=2}^{g(i,k)} (D_{itk} - D_{i(t-1)k}) h_{itk} = \sum_{j=1}^{N} d_{ijk} w_{ijk}$$

convirtiéndose el problema en el siguiente:

$$(P_3) \begin{cases} \text{Mín} & \displaystyle\sum_{i=1}^{N} \sum_{k=1}^{m} \sum_{t=2}^{g(i,k)} (D_{itk} - D_{i(t-1)k}) h_{itk} & \\ \text{s.a:} & \displaystyle\sum_{j=1}^{r} x_{ij} = 1 & \forall i \\ & \displaystyle\sum_{k=1}^{m} z_k = q & \\ & h_{itk} + \displaystyle\sum_{\{j/d_{ijk} < D_{itk}\}} x_{ij} \geq z_k & \forall i, \forall k, \forall t \geq 2 \\ & h_{itk} \geq 0 & \forall i, \forall k, \forall t \\ & x_{ij} \geq 0 & \forall i, \forall j \\ & z_k \in \{0, 1\} & \forall k \end{cases}$$

La restricción $h_{itk} + \displaystyle\sum_{\{j/d_{ijk} < D_{itk}\}} x_{ij} \geq z_k \quad \forall i, \forall k, \forall t \geq 2$ asegura que una variable radial h_{itk} debe tomar el valor 1 si $z_k = 1$ y $x_{ij} = 0$ para todo j que esté más cerca que la distancia D_{itk}. Como se está minimizando y las distancias son positivas, existe una solución óptima donde todas las variables x_{ij} son binarias. No es necesario imponer que h_{itk} sean binarias, pues tomarán valor 0 ó 1 en cualquier solución óptima.

3.3.5. Métodos Heurísticos

La complejidad del problema radica en el hecho de que se calculan de forma simultánea las variables óptimas z y las asignaciones óptimas x. Sin embargo, cuando una de las dos es fijada previamente, el problema resultante es resoluble en tiempo polinomial. Fijado z, se calculan las distancias de O_i a R_j y se procede a la asignación de cada O_i al R_j más cercano. Por otro lado, en la prueba del teorema 3.3 se muestra que se puede calcular un óptimo en z estableciendo un orden en las distancias. El método descrito a continuación alterna entre encontrar un óptimo en z fijadas unas asignaciones x y viceversa, hasta alcanzar un óptimo local.

Subrutina de mejores asignaciones:

- Entrada: El subconjunto $Q \subseteq V$.
- Salida: La matriz de asignaciones X y la función objetivo $D_Q(X)$.
- Paso 1: Para todo $O_i \in O$ y $R_j \in R$, se define $c_{ij} = \sum_{v_k \in Q} d_{ijk}$
- Paso 2: Para todo $O_i \in O$, sea $x_{ij} = 1 \iff c_{ij} = \text{mín}\{c_{ik} | 1 \leq k \leq r\}$; $x_{ij} = 0$ c.c.
- Paso 3: Se define $D_Q(X) = \sum_{O_i \in O} \sum_{R_j \in R} c_{ij} x_{ij}$

El número de operaciones involucradas es $Nmr + Nr + Nr$, por lo que en el peor de los casos la complejidad puede llegar a ser cúbica (O(Nrm)). La siguiente subrutina calcula un subconjunto óptimo Q para unas asignaciones X de clusters dadas.

Subrutina de mejores variables:

- Entrada: Matriz de asignaciones X.
- Salida: subconjunto $Q \subseteq V$ de variables y función objetivo $D_X(Q)$.
- Paso 1: Para todo $k \in V$, se define $b_k = \sum_{i,j} d_{ijk} x_{ij}$
- Paso 2: Ordenar b_j en orden creciente para obtener: $b_{j(1)} \leq ... \leq b_{j(k)}$
- Paso 3: $v_{j(i)} \in Q \iff i \leq q$
- Paso 4: Se define $D_X(Q) = \sum_{i=1}^{q} b_{j(i)}$

En este caso, el número de operaciones involucradas es $Nrm + mlog(m) + q$. Por tanto, en el peor de los casos también se tiene complejidad cúbica O(Nrm).

Estas dos subrutinas se pueden combinar en el siguiente algoritmo:

Algoritmo q-VarSel

- Paso 1: Seleccionar aleatoriamente un conjunto de variables Q^0 y hacer $t := 0$.
- Repetir hasta que se logre un óptimo local: $D_{Q^t}(X^t) = D_{X^t}(Q^{t+1})$.
  - Paso 2: Asignación de unidades: Para Q^t dado, llamar a la subrutina de mejores asignaciones para calcular X^t óptimo y $D_{Q^t}(X^t)$.
  - Paso 3: Selección de variables: Para X^t dado, llamar a la subrutina de mejores variables para calcular Q^{t+1} óptimo y $D_{X^t}(Q^{t+1})$. Después, actualizar $t := t + 1$.
- Paso 4: Si $t \leq t^{max}$ volver al paso 1.

Por construcción, es fácil ver que $D_{Q^t}(X^t) \geq D_{X^t}(Q^{t+1})$. Como hay un número finito de conjuntos Q que pueden ser elegidos, tarde o temprano se logrará $D_{Q^t}(X^t) = D_{X^t}(Q^{t+1})$, alcanzando así el óptimo local. Este proceso se ejecuta un total de t^{max} veces (valor prefijado).

La calidad de este algoritmo se puede comparar con métodos heurísticos alternativos para el problema. Por ejemplo, se puede utilizar la subrutina de mejores asignaciones para actualizar Q^t mediante un intercambio de variables ($v_j \in Q^t$ por $v_i \notin Q^t$) como sucede en el siguiente algoritmo:

Algoritmo Add-Drop:

- Paso 1: Seleccionar de forma aleatoria un conjunto de variables Q^0 y hacer $t := 0$.
- Repetir hasta que se alcance un óptimo local: $D(Q^t) = D(Q^{t+1})$.
  - Paso 2: Añadir: Calcular $D(Q^t \bigcup \{v_{i^*}\}) = \min_{v_i \notin Q^t} D(Q^t \bigcup \{v_i\})$
  - Paso 3: Expulsar: Calcular $D(Q^t \bigcup \{v_{i^*}\} - \{v_{j^*}\}) = \min_{v_j \in Q^t; v_j \neq v_{i^*}} D(Q^t \bigcup \{v_{i^*}\} - \{v_j\})$. Actualizar $Q^{t+1} = Q^t \bigcup \{v_{i^*}\} - \{v_{j^*}\}$ y $t := t + 1$.
- Paso 4: Si $t \leq t^{max}$, volver al paso 1.

Como puede verse en el proceso, este algoritmo mejora una solución Q^t a una nueva Q^{t+1} mediante la adición o expulsión de variables. El nuevo valor Q^{t+1} se calcula después de la evaluación de m funciones objetivo. El intercambio completo de pares de variables puede incrementar dicho término lineal a uno cuadrático (m^2), haciéndolo menos práctico en situaciones donde se tengan grandes bases de datos. Sin embargo, el algoritmo Add-Drop constituye la piedra angular del enfoque más satisfactorio al problema de la p-mediana.

4. Comparativa de modelos

Los modelos lineal y radial correspondientes a la sección 3.2 fueron implementados para probar su rendimiento en distintas bases de datos. Dichas bases se distinguen en dos grupos:

- Un primer grupo utilizando valores procedentes de una distribución $U(0, 7000)$ para aquellos datos de menor tamaño: valores N comprendidos entre 10 y 25 y m comprendidos entre 3 y 5, con los que se hicieron variar M y q entre 2 y 4.
- Un segundo grupo en el que los datos se generaron mediante el mismo método de generación que el utilizado en [11] por los autores Michael J. Brusco y Douglas Steinley, con los que se realizaron pruebas para tamaños similares y mayores a los anteriores: valores N comprendidos entre 25 y 40, m entre 5 y 8 haciendo variar M y q entre 4 y 10.

Para cada dos valores fijos de N y m, se generaron 5 bases de datos según uno de los criterios anteriores. A continuación, se ejecutaron ambos modelos en las 5 bases de datos para otros valores fijos de M y q. Se tomó el promedio obtenido en la ejecución de estos 5 ficheros para los valores M y q con que fueron ejecutados en cada modelo por separado. Los resultados obtenidos muestran que para un tamaño de datos pequeño el modelo lineal obtiene solución óptima en un tiempo ligeramente menor al modelo radial. Sin embargo, según se hace aumentar el tamaño de los datos y se hace variar el resto de parámetros, aumenta el tiempo en que el modelo lineal obtiene dicho óptimo, llegando a superar los 30 minutos de ejecución sin obtener solución en las bases de mayor tamaño. Por contra, el modelo radial mantiene su rendimiento cuando se hace aumentar el tamaño de datos, no superando los 5 minutos de ejecución salvo para las bases de mayor tamaño. Esto lleva a afirmar, como ya se adelantó al introducir esta sección, que el modelo basado en la formulación radial del problema resulta el mejor, en el sentido de que obtiene el óptimo para el problema en un tiempo no excesivamente alto en comparación al modelo lineal. En base a estos datos, se puede concluir que el modelo basado en formulación radial resulta más eficiente a la hora de abordar casos en los que se trabaje con bases de datos de gran tamaño. Dichos resultados pueden observarse en el anexo a este trabajo, donde se muestran los tiempos promedio de ejecución para ambos modelos siguiendo el procedimiento explicado. La columna N indica el número de individuos que se tomaron para la prueba, m indica el número de variables iniciales con respecto a las cuales se toma información de

los N individuos. Las columnas M y q indican los clusters a obtener en la clasificación y el número de variables con el que se realiza la misma, respectivamente. Las columnas Lineal y Radial muestran los tiempos medios de ejecución de cada modelo. Cuando se superan los 30 minutos de ejecución en más de una de las 5 bases con que se trabaja para cada caso sin obtener una solución óptima se hace indicar en dichas columnas mediante *.

Las pruebas fueron realizadas en un equipo Windows 64-bits Intel Celeron P4600 con dos procesadores de 2.00 GHz y 3.00 GB RAM. Los códigos de ambos modelos fueron escritos en Xpress y se encuentran disponibles en el anexo.

5. Conclusiones

En el presente trabajo se ha presentado el Análisis Cluster (Análisis de Conglomerados) como método para la clasificación de individuos mediante un punto de vista basado en la programación matemática, así como algoritmos para obtener distintas particiones (basados en teoría de grafos, programación dinámica, branch and bound o heurísticos), ya sean resolubles en tiempo polinomial cuando es posible, o algoritmos no resolubles en tiempo polinomial que resultan útiles aplicados a problemas NP duros.

Así mismo, la selección de variables también ha sido expuesta como un proceso previo importante en la detección de variables no necesarias en el análisis, presentándola con un modelo en programación no lineal y siendo linealizada a modelos en programación lineal entera mixta para la clasificiación y selección simultáneas. Se han realizado pruebas a dichos modelos indicando la velocidad media de ejecución de cada uno para ciertos tamaños de datos, mostrando el mejor rendimiento en el modelo basado en formulación radial (sobre todo al aumentar el tamaño de los datos con que se trabaja). Se ha tratado también el caso en el que son prefijados unos centros, mostrando que incluso reduciendo el problema de esta forma resulta NP completo, por lo que los métodos heurísticos propuestos pueden ser útiles a la hora de abordar este problema.

En general, la selección de variables aporta mayor robustez a la clasificiación debido a que descarta aquellas variables que influyen en el análisis de forma innecesaria, proporcionando soluciones más precisas del mismo, y en consecuencia, una clasificación más acertada.

A. Anexo

A.1. Tablas de pruebas

N	m	M	q	Lineal	Radial
10	3	2	2	0.38s	0.56s
10	4	2	2	0.76s	0.85s
10	4	2	3	0.79s	1.05s
10	4	3	3	0.94s	1.05s
10	5	2	2	0.91s	1.19s
10	5	2	3	1.51s	1.22s
10	5	3	3	1.55s	1.17s
10	5	3	4	1.02s	1.2s
10	5	4	4	1.79s	0.95s
15	3	2	2	1.24s	1.41s
15	4	2	2	1.84s	1.97s
15	4	2	3	2.35s	1.74s
15	4	3	3	2.74s	2.03s
15	5	2	2	3.09s	3.16s
15	5	2	3	4.55s	2.89s
15	5	3	3	4.6s	2.72s
15	5	3	4	8.92s	2.31s
15	5	4	4	9.3s	2.12s
20	3	2	2	2.53s	3.05s
20	4	2	2	4.27s	3.8s
20	4	2	3	5.64s	3.87s
20	4	3	3	3.71s	6.6s
20	5	2	2	9.4s	6.48s
20	5	2	3	7.12s	5.67s
20	5	3	3	37.4s	5.4s
20	5	3	4	27.6s	4.56s
20	5	4	4	2min 27s	3.5s

N	m	M	q	Lineal	Radial
25	3	2	2	3.05s	4.7s
25	4	2	2	10.4s	7.22s
25	4	2	3	10.7s	6.34s
25	4	3	3	1min 13s	6.53s
25	5	2	2	22s	11.2s
25	5	2	3	23.6s	11.32s
25	5	3	3	23.30s	12s
25	5	3	4	1min 35s	7.64s
25	5	4	4	1min 13s	6.4s
25	8	4	4	5min 20s	38s
25	8	4	6	*	38.2s
25	8	6	6	*	14.8s
35	8	6	4	*	4min 50s
35	8	6	6	*	2min
35	8	8	4	*	4min 57s
35	8	8	6	*	42.23s
35	8	10	4	*	11min 12s
35	8	10	6	*	44.16s
40	8	4	4	*	4min 16s
40	8	4	6	*	2min 21s
40	8	6	4	*	13min 23s
40	8	6	6	*	1min 15s
40	8	8	4	*	10min 45s
40	8	8	6	*	1min 20s

A.2. Códigos utilizados para los modelos (Xpress)

```
model Modelo Lineal
uses "mmxprs";
uses "mmsystem";

parameters
M=25
N=5
p=4
q=4
end-parameters

declarations
Entidades=1..M
Medianas=1..M
Variables=1..N
tiempoinicial: real
tiempofinal: real

d: array(Entidades, Entidades,Variables) of real
x: array(Entidades, Medianas) of mpvar
y: array(Medianas) of mpvar
z: array(Variables) of mpvar
mindif: mpvar

Datos: array(Entidades, Variables) of real
w: array(Entidades, Medianas,Variables) of mpvar
end-declarations

fopen("Datos25x5.txt",F_INPUT)
forall(i in 1..M)do
        forall(k in 1..N)read(Datos(i,k))
        readln
end-do
fclose(F_INPUT)

forall (i in Entidades, j in Entidades, k in Variables)
   d(i,j,k):= abs(Datos(i,k)-Datos(j,k))


mindif= sum(i in 1..M,j in 1..M,k in 1..N) d(i,j,k)*w(i,j ,k)

forall(i in 1..M, j in 1..M) x(i,j)<=y(j)
forall (i in 1..M) sum(j in 1..M) x(i,j)=1
```

```
sum(j in 1..M) y(j)=p
sum(k in 1..N) z(k)=q
forall(i in 1..M, j in 1..M, k in 1..N) w(i,j,k)>=x(i,j)+z(k)-1
forall(i in 1..M, j in 1..M, k in 1..N) w(i,j,k)<=x(i,j)
forall(i in 1..M, j in 1..M, k in 1..N) w(i,j,k)<=z(k)
forall(j in 1..M) y(j) is_binary
forall(k in 1..N) z(k) is_binary

tiempoinicial:=gettime

minimize(mindif)

tiempofinal:=gettime

writeln("Soluci n:\n Valor objetivo: ", getobjval)
writeln("Asignaciones realizadas: ")
forall(i in 1..M,j in 1..M) do
        if (getsol(x(i,j))<>0) then
                writeln("x(",i,",",j,")=",getsol(x(i,j)))
        end-if
end-do

writeln

writeln("Medianas seleccionadas: ")
forall(j in 1..M) do
        if (getsol(y(j))<>0) then
                writeln("y(",j,")=",getsol(y(j)))
        end-if
end-do

writeln

writeln("Variables seleccionadas: ")
forall(k in 1..N) do
        if (getsol(z(k))<>0) then

                writeln("z(",k,")=",getsol(z(k)))
        end-if
end-do


writeln("Tiempo de computaci n \n Modelo lineal: ",tiempofinal-
    tiempoinicial)
end-model
```

```
model Modelo Radial
uses "mmxprs";
uses "mmsystem";
parameters
M=25
N=3
p=2
q=2
end-parameters

declarations
Entidades=1..M
Medianas=1..M
Variables=1..N
tiempoinicial: real
tiempofinal: real

C:array(1..M,1..N)of real
D:array(1..M,1..M,1..N) of real
c:array(set of integer) of real
ci:array(1..M,1..N,0..M) of real     !i,k,j
Gi:array(1..M,set of integer) of integer      !i,k
G:integer
end-declarations


fopen("Datos25x3.txt",F_INPUT)
forall(i in 1..M)do
        forall(j in 1..N)read(C(i,j))
        readln
end-do
fclose(F_INPUT)


forall (i in 1..M, j in 1..M,k in 1..N)
    D(i,j,k):=abs(C(i,k)-C(j,k))


c(0):=0
h:=0
repeat
        h:=h+1
        c(h):=min(i in 1..M,j in 1..M,k in 1..N|D(i,j,k)>c(h-1))
            D(i,j,k)
until(c(h)=max(i in 1..M,j in 1..M,k in 1..N)D(i,j,k))
```

```
G:=h


forall(i in 1..M,k in 1..N) do
    ci(i,k,0):=0
        h:=0
        repeat
                h:=h+1
                ci(i,k,h):=min(j in 1..M|D(i,j,k)>ci(i,k,h-1))D(
                    i,j,k)
                writeln(ci(i,k,h))
        until(ci(i,k,h)=max(j in 1..M)D(i,j,k))
        Gi(i,k):=h
end-do

declarations
        x:array(1..M,1..M)of mpvar
        z: array(1..N) of mpvar
        r: array(1..M,1..N,2..G) of mpvar
        y: array(1..M) of mpvar
end-declarations


forall(k in 1..N)z(k) is_binary
forall(j in 1..M)y(j) is_binary
forall(i in 1..M, k in 1..N, t in 2..Gi(i,k)) r(i,k,t)+sum(j in
    1..M|D(i,j,k)<ci(i,k,t))x(i,j)>=z(k)
forall(i in 1..M,j in 1..M) x(i,j)<= y(j)
forall (i in 1..M) sum(j in  1..M)x(i,j)=1
sum(j in 1..M)y(j)=p
sum(k in 1..N) z(k)=q


tiempoinicial:=gettime

minimize(sum(i in 1..M,k in 1..N,t in 2..Gi(i,k))(ci(i,k,t)-ci(i
    ,k,t-1))*r(i,k,t))

tiempofinal:=gettime


writeln("Soluci n:\n Valor objetivo: ", getobjval)
writeln("Asignaciones realizadas: ")
forall(i in 1..M,j in 1..M) do
        if (getsol(x(i,j))<>0) then
```

```
                writeln("x(",i,",",j,")=",getsol(x(i,j)))
        end-if
end-do

writeln

writeln("Medianas seleccionadas: ")
forall(j in 1..M) do
        if (getsol(y(j))<>0) then
                writeln("y(",j,")=",getsol(y(j)))
        end-if
end-do

writeln

writeln("Variables seleccionadas: ")
forall(k in 1..N) do
        if (getsol(z(k))<>0) then

                writeln("z(",k,")=",getsol(z(k)))
        end-if
end-do

writeln

writeln("Tiempo de computaci n \n Modelo radial: ",tiempofinal-
    tiempoinicial)
end-model
```

Referencias

[1] HANSEN, P. y JAUMARD, B., (1997) Cluster analysis and mathematical programming, *Mathematical Programming* **79**, 191-215.

[2] BENATI, S., GARCÍA, S. y PUERTO, J., *Optimization Methods to Select Variables for Clustering*.
Preprint submitted for publication.

[3] BENATI, S. y GARCÍA, S., (2014) A Mixed Integer Linear Model for Clustering with Variable Selection, *Computers & Operations Research* **43**, 280-285.

[4] ROSENSTIEHL, P., (1967), L'arbre minimum d'un graphe, *Théorie des Graphes*, 357-368.

[5] DELATTRE, M. y HANSEN, P., (1980), *Bicriterion cluster analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2**, 277-291.

[6] GUÉNOCHE, A., (1989), Partitions with minimum diameter, *Proceedings of the International Conference of the Federation of Classification Societies*

[7] MONMA, C. y SURI, S., (1991), Partitioning points and graphs to minimize the maximum or the sum of diameters, *Proceedings of the Sixth Quadrennial International Conference on the Theory and Applications of Graphs, Graphs Theory, Combinatorics, and Applications*, 899-912.

[8] HANSEN, P., JAUMARD, B. y FRANK, O., (1989), Maximum sum-of-splits clustering, *J. Classification 6*, 177-193.

[9] CHRISTOFIDES, N., (1975), *Graph Theory. An Algoritmic Approach.*

[10] HANSEN, P. y JAUMARD, B., (1987), Minimum sum of diameters clustering, *J. Classification* **4**, 215-226.

[11] STEINLEY, D. y BRUSCO, M., (2008), Selection of variables in cluster analysis: An empirical comparison of eight procedures, *Psychometrica* **73**, 125-144.